

Fig. 1
Prior Art

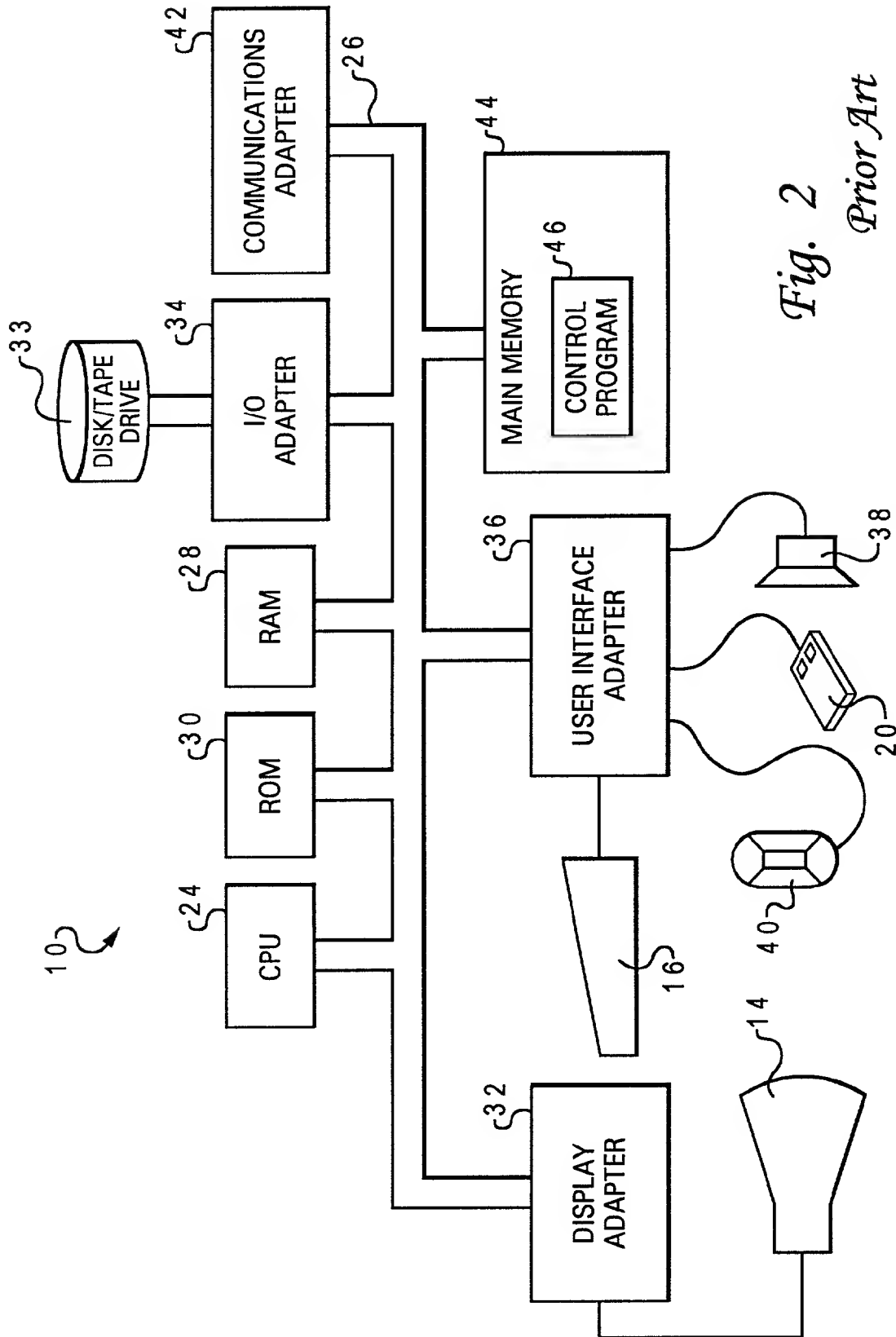


Fig. 2
Prior Art

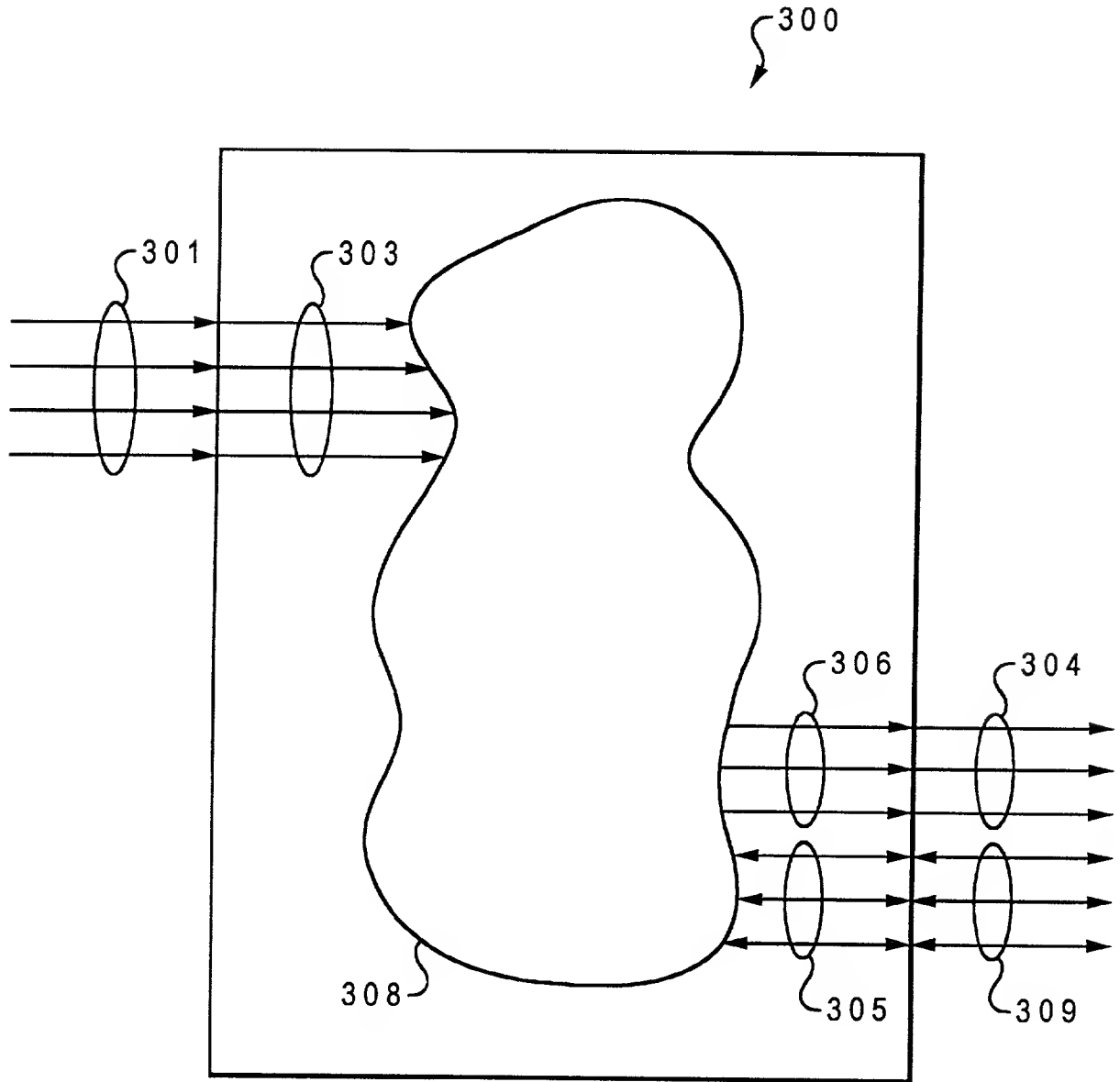


Fig. 3A

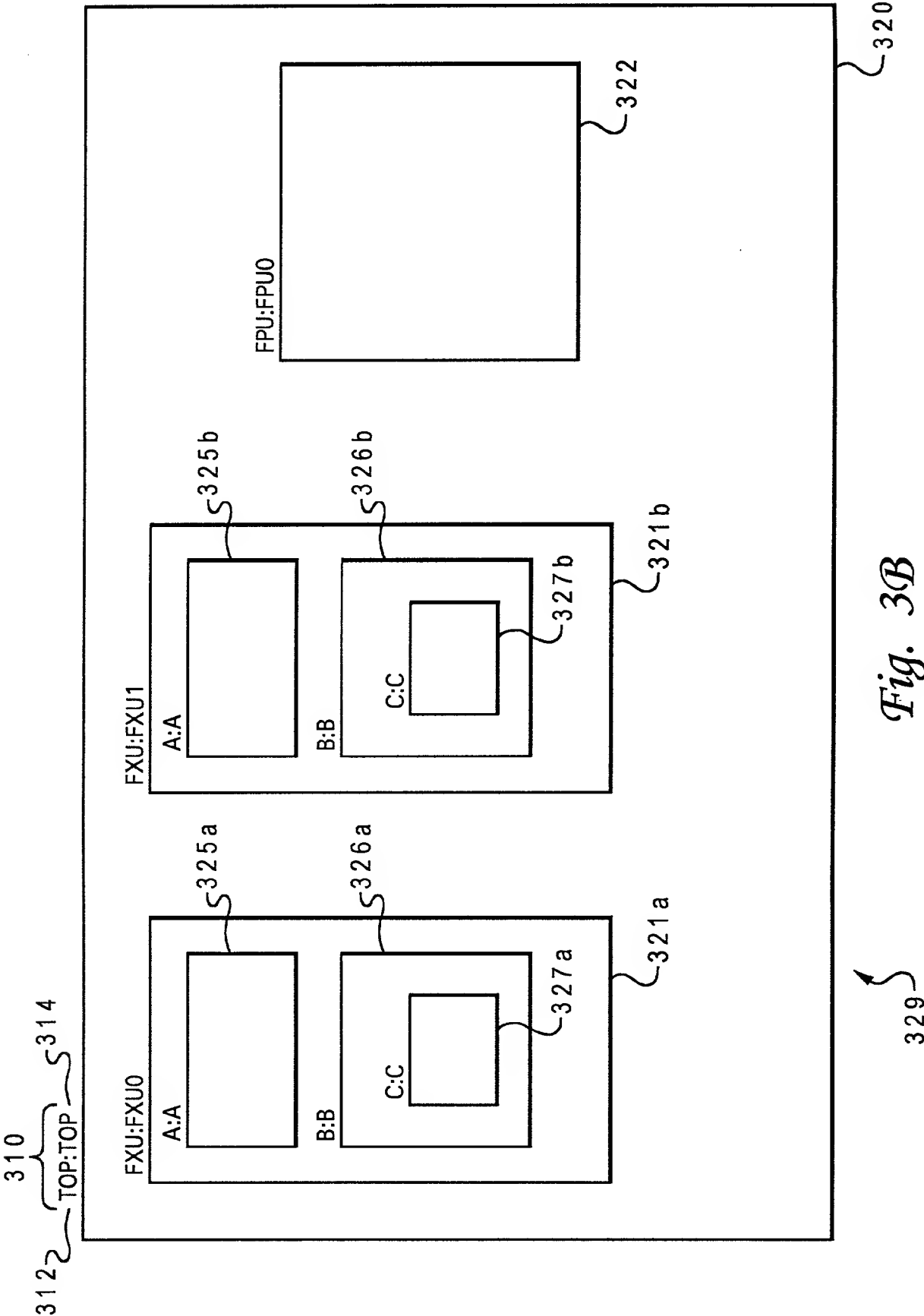


Fig. 3B

5/62

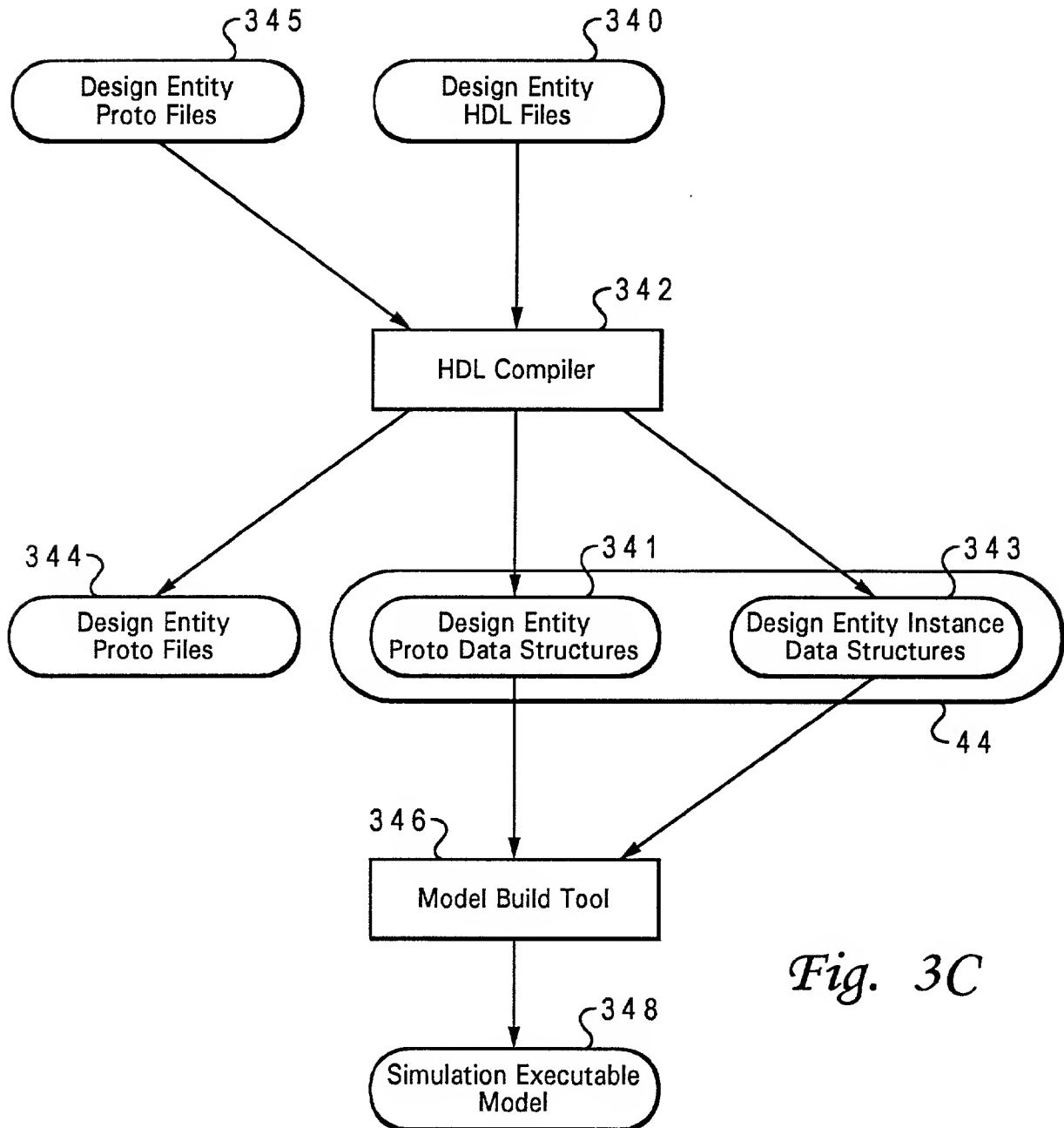


Fig. 3C

6/62

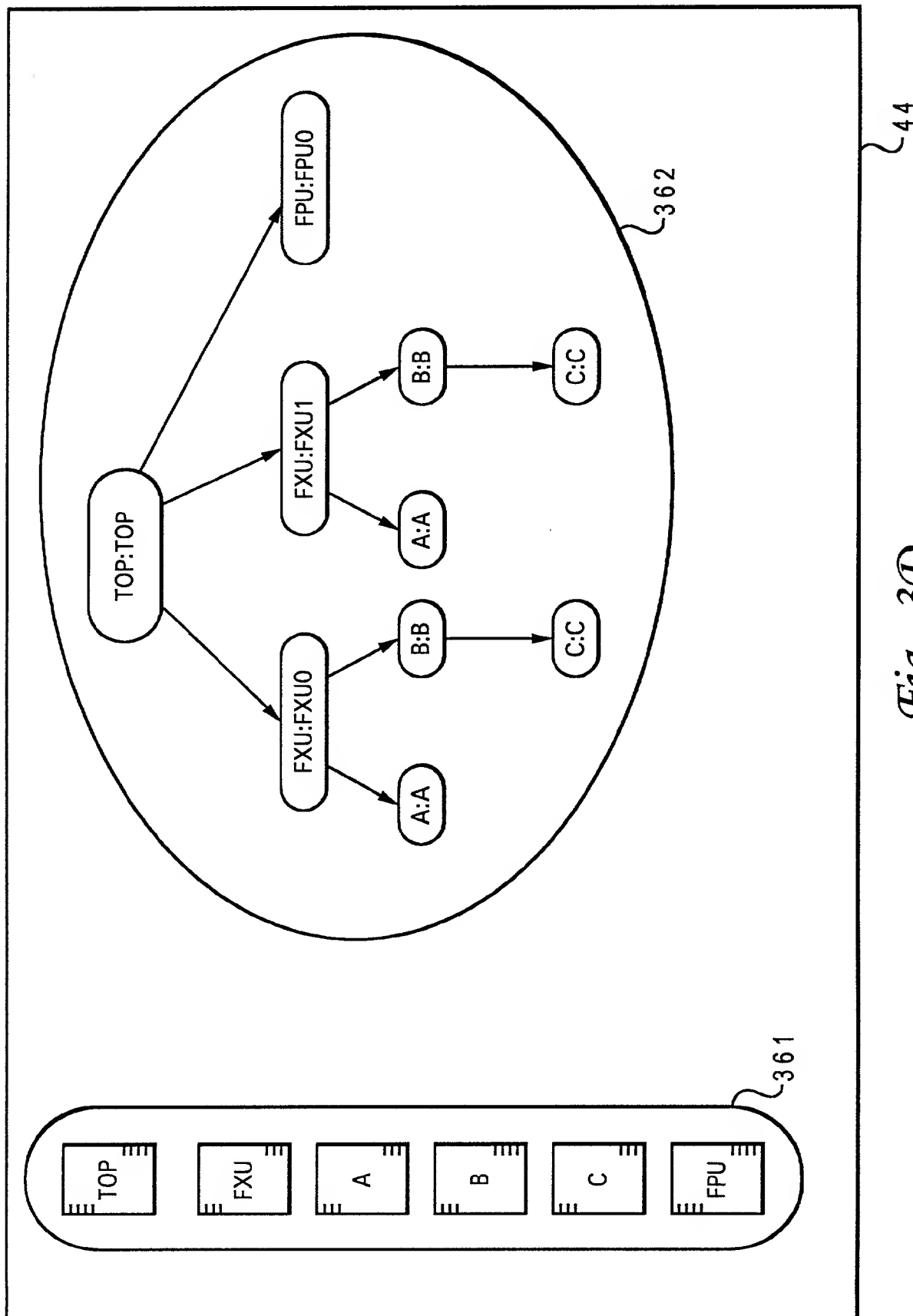


Fig. 3D

7/62

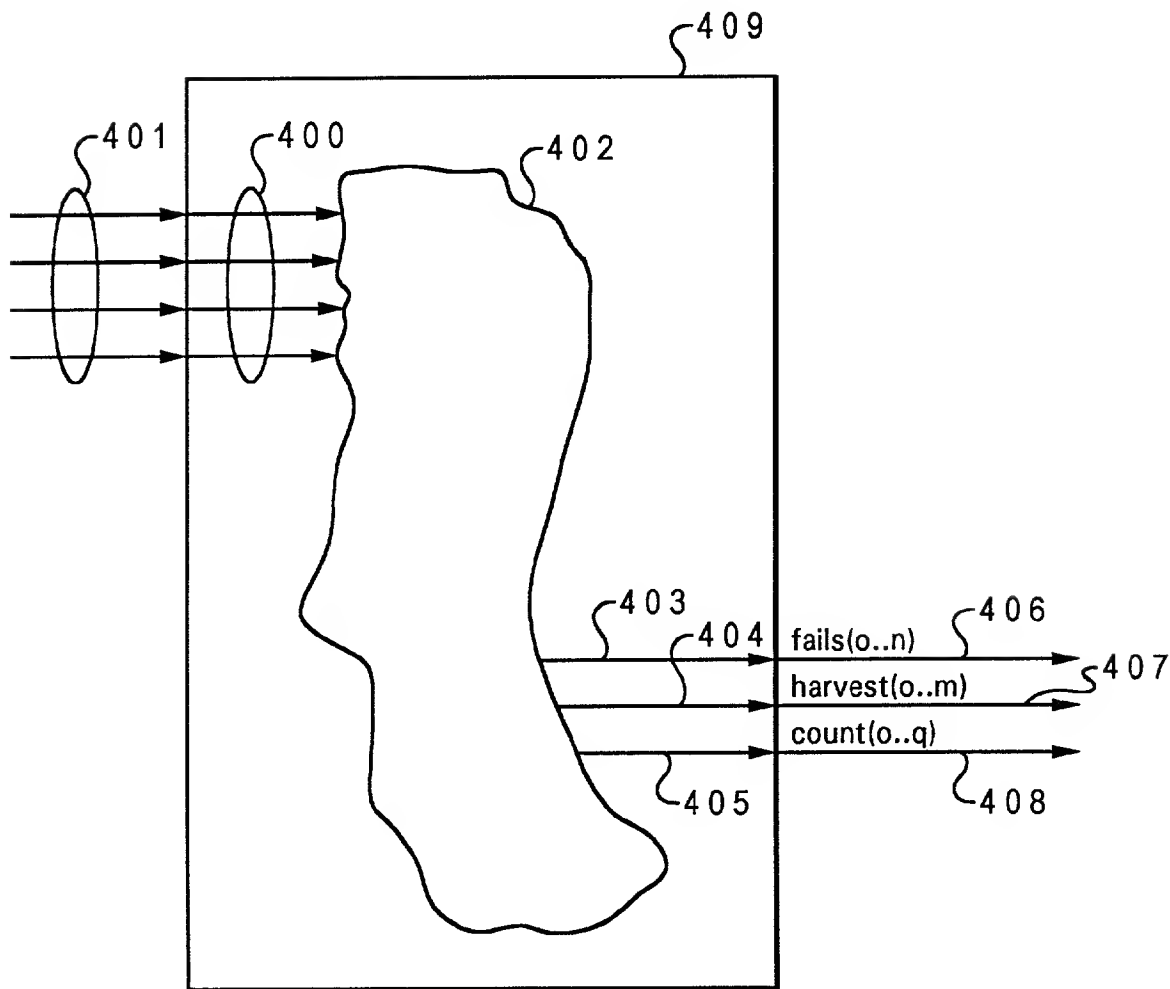


Fig. 4A

8/62

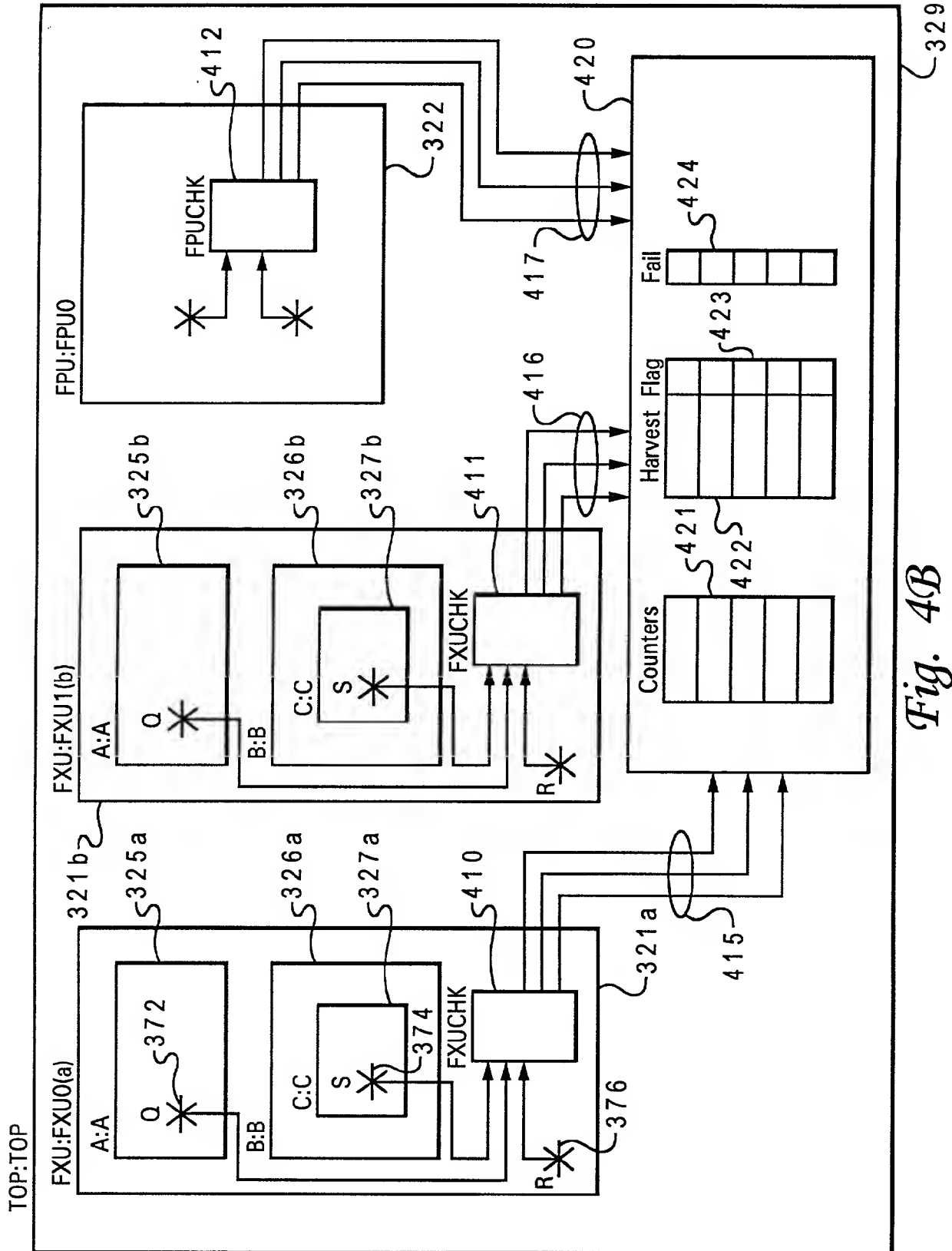


Fig. 4B

20220820 20220820

9/62

ENTITY FXUCHK IS

```

PORT(  S_IN      :    IN std_ulogic;
        Q_IN      :    IN std_ulogic;
        R_IN      :    IN std_ulogic;
        clock     :    IN std_ulogic;
        fails     :    OUT std_ulogic_vector(0 to 1);
        counts    :    OUT std_ulogic_vector(0 to 2);
        harvests  :    OUT std_ulogic_vector(0 to 1);
);

```

4 5 0

4 5 2 { --!! BEGIN
--!! Design Entity: FXU;

4 5 3 { --!! Inputs
--!! S_IN => B.C.S;
--!! Q_IN => A.Q;
--!! R_IN => R;
--!! CLOCK => clock;
--!! End Inputs

4 5 4 { --!! Fail Outputs;
--!! 0 : "Fail message for failure event 0";
--!! 1 : "Fail message for failure event 1";
--!! End Fail Outputs;

4 5 5 { --!! Count Outputs;
--!! 0 : <event0> clock;
--!! 1 : <event1> clock;
--!! 2 : <event2> clock;
--!! End Count Outputs;

4 5 6 { --!! Harvest Outputs;
--!! 0 : "Message for harvest event 0";
--!! 1 : "Message for harvest event 1";
--!! End Harvest Outputs;

4 5 7 { --!! End;

4 5 1

4 4 0

ARCHITECTURE example of FXUCHK IS

BEGIN

... HDL code for entity body section ...

END;

4 5 8

Fig. 4C

10/62

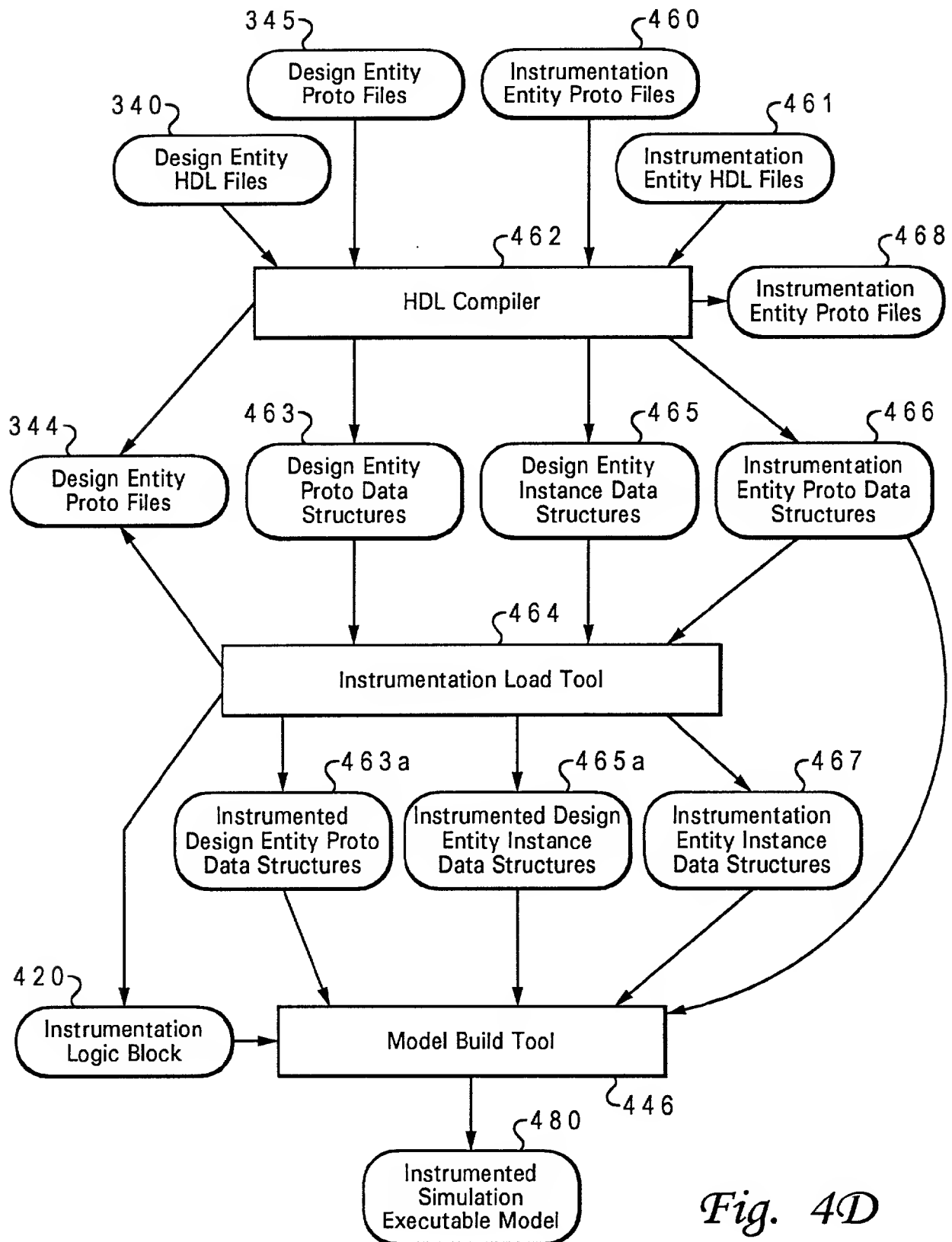


Fig. 4D

11/62

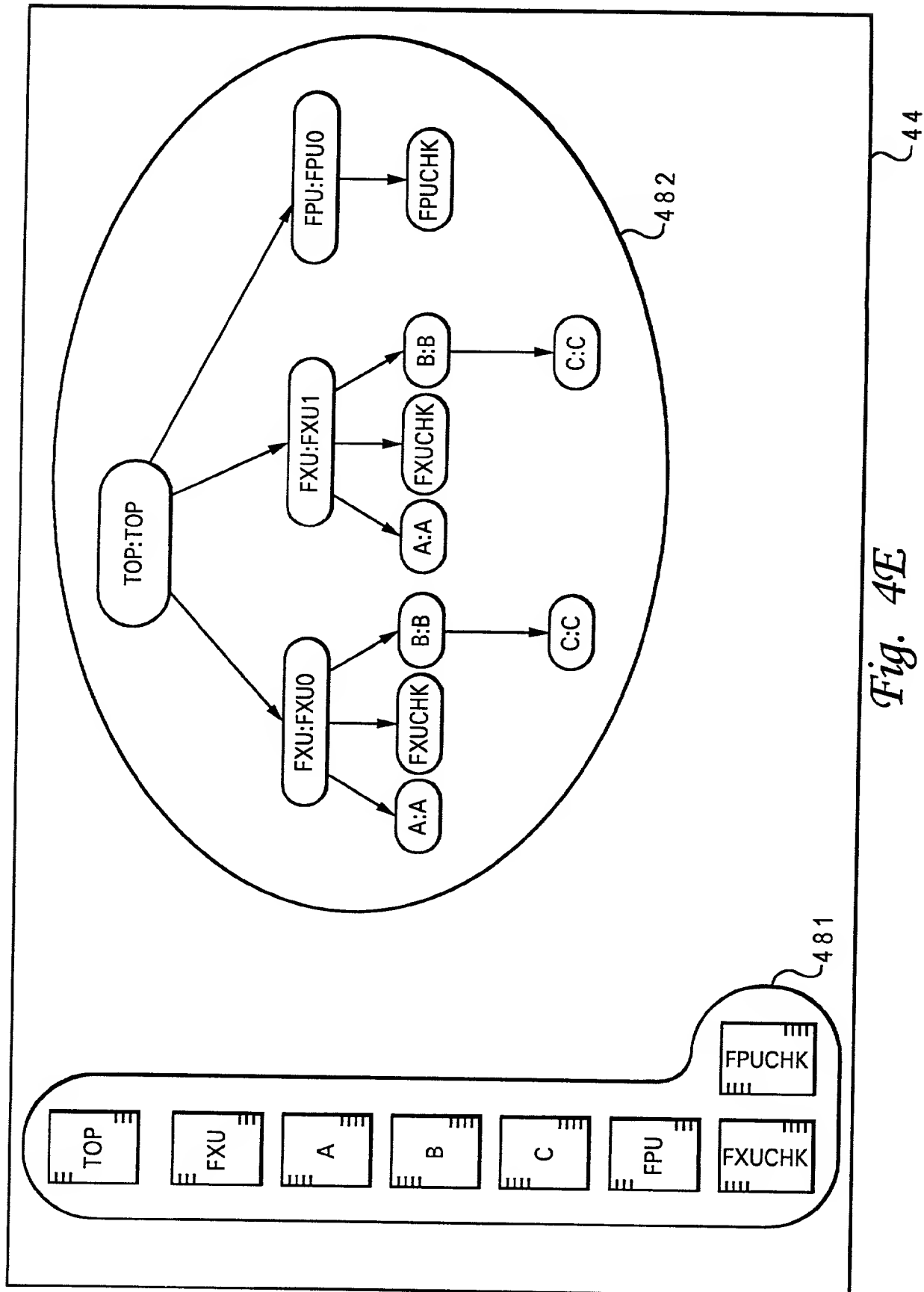
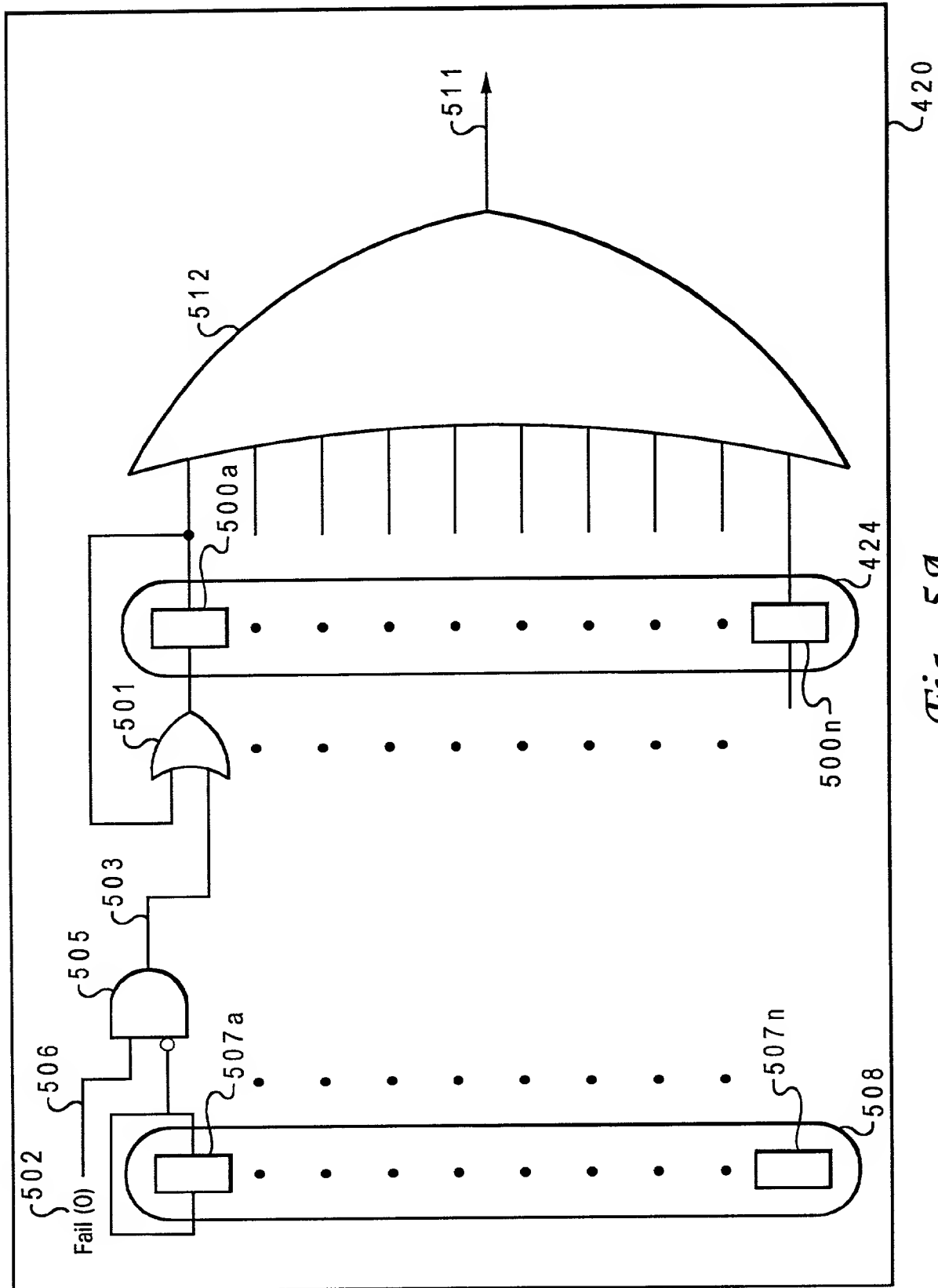


Fig. 4E



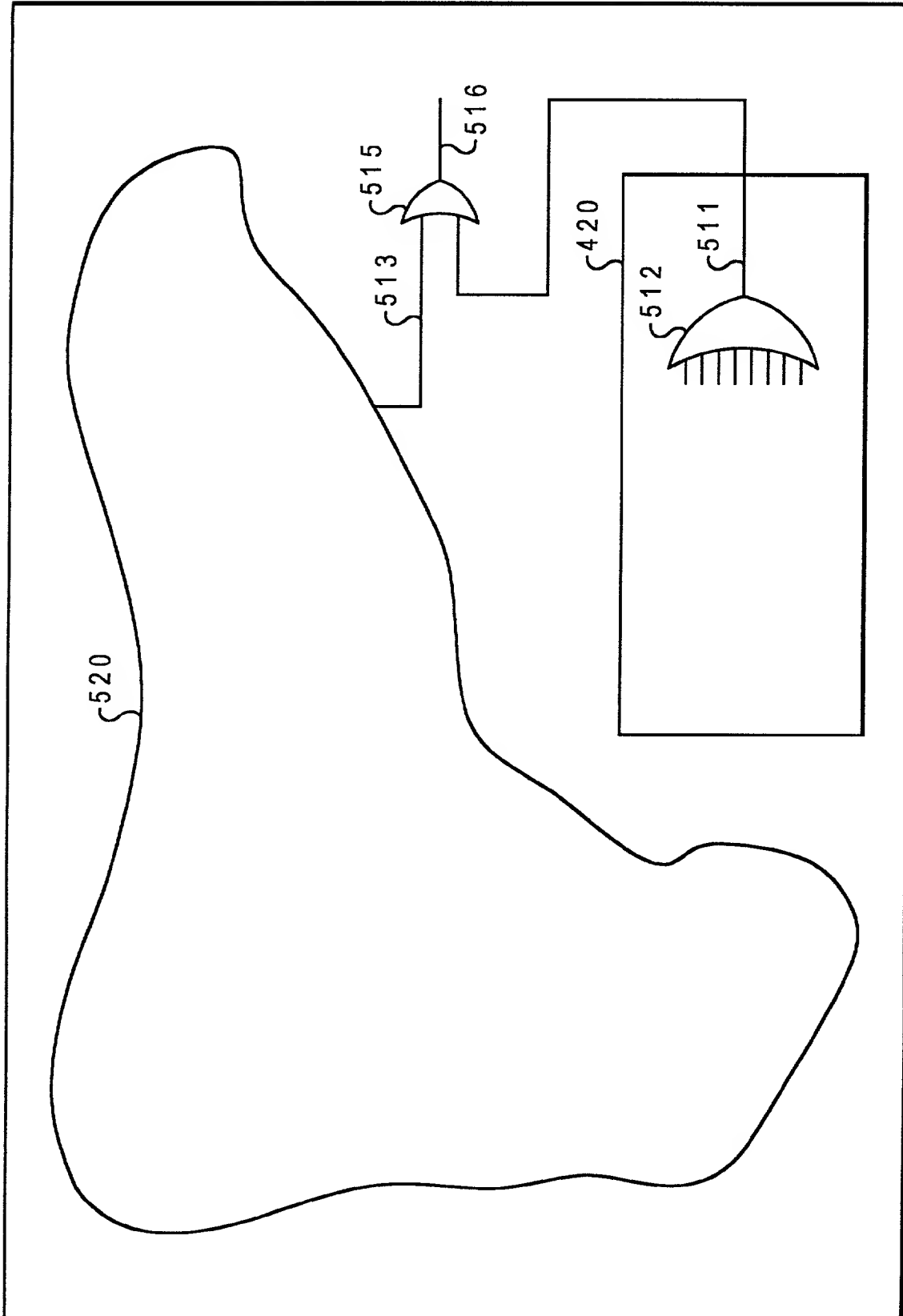
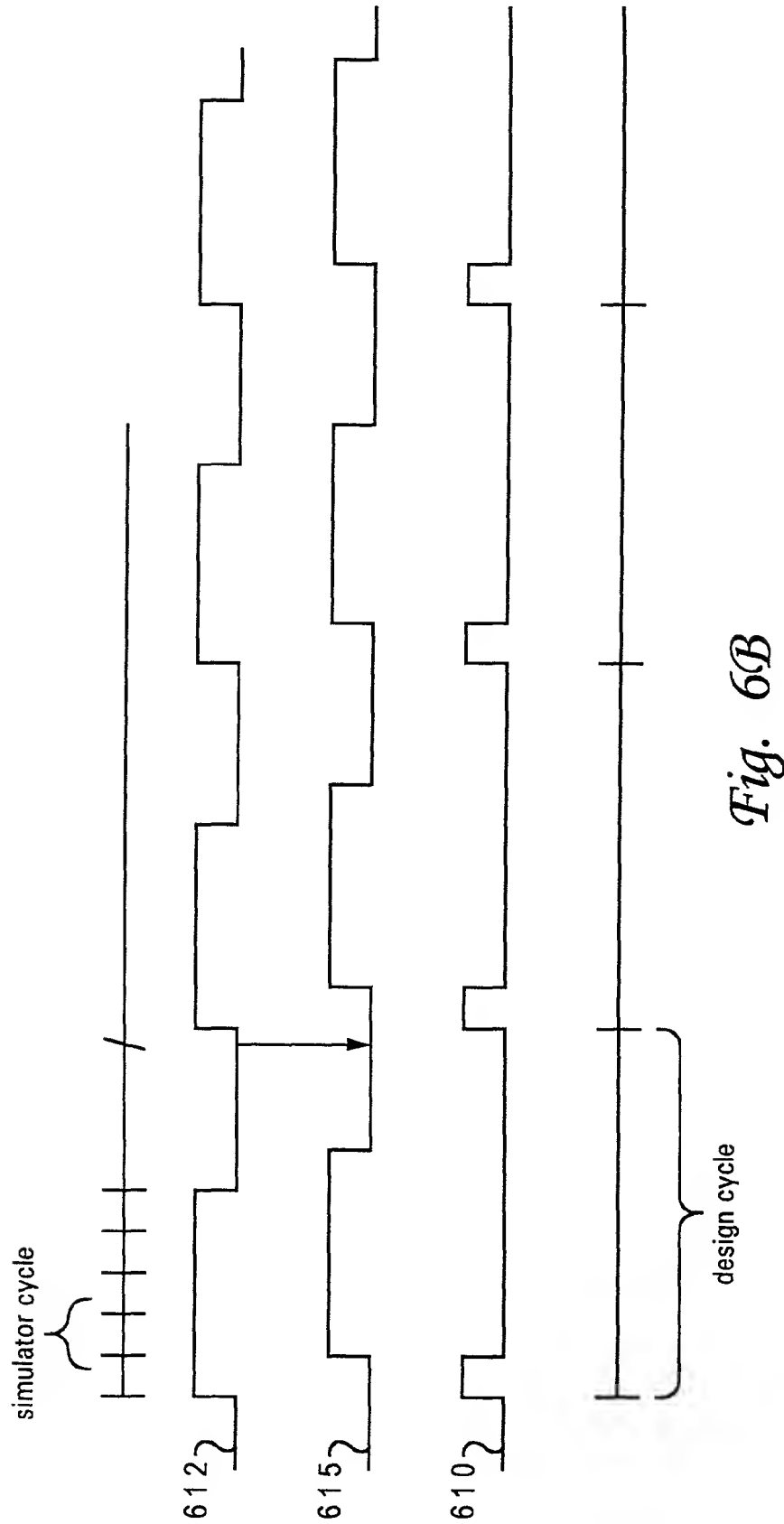


Fig. 5B



15/62



16/62

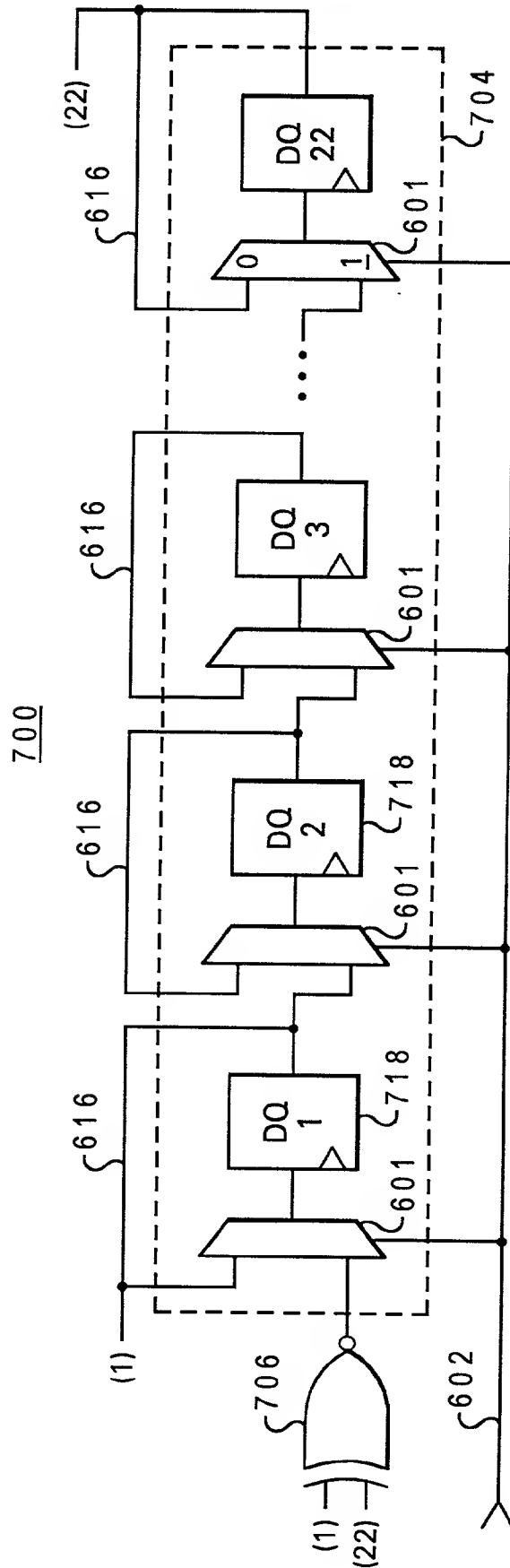


Fig. 7

17/62

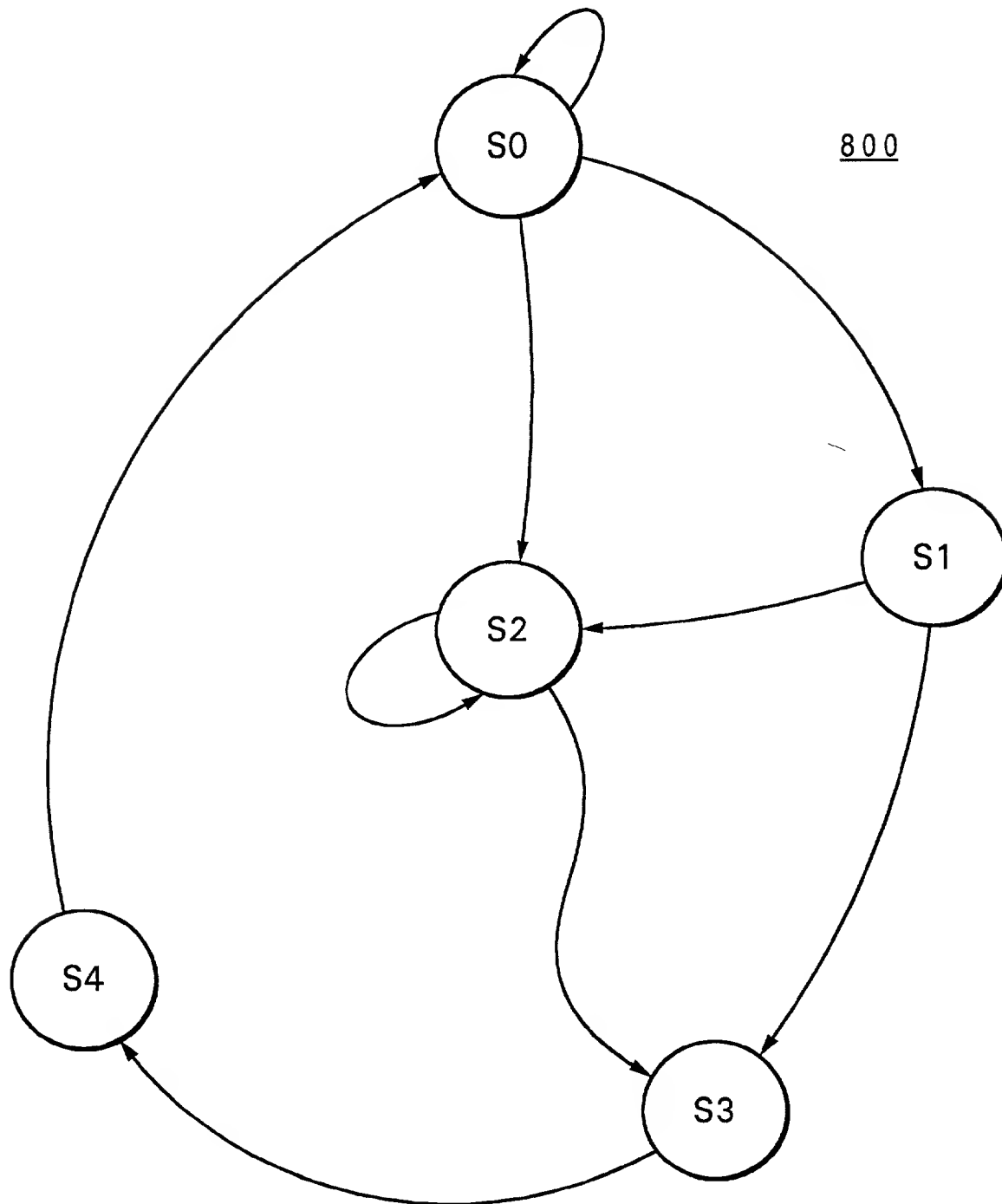


Fig. 8A
Prior Art

09997802-023409
2012220-20826660

18/62

entity FSM : FSM

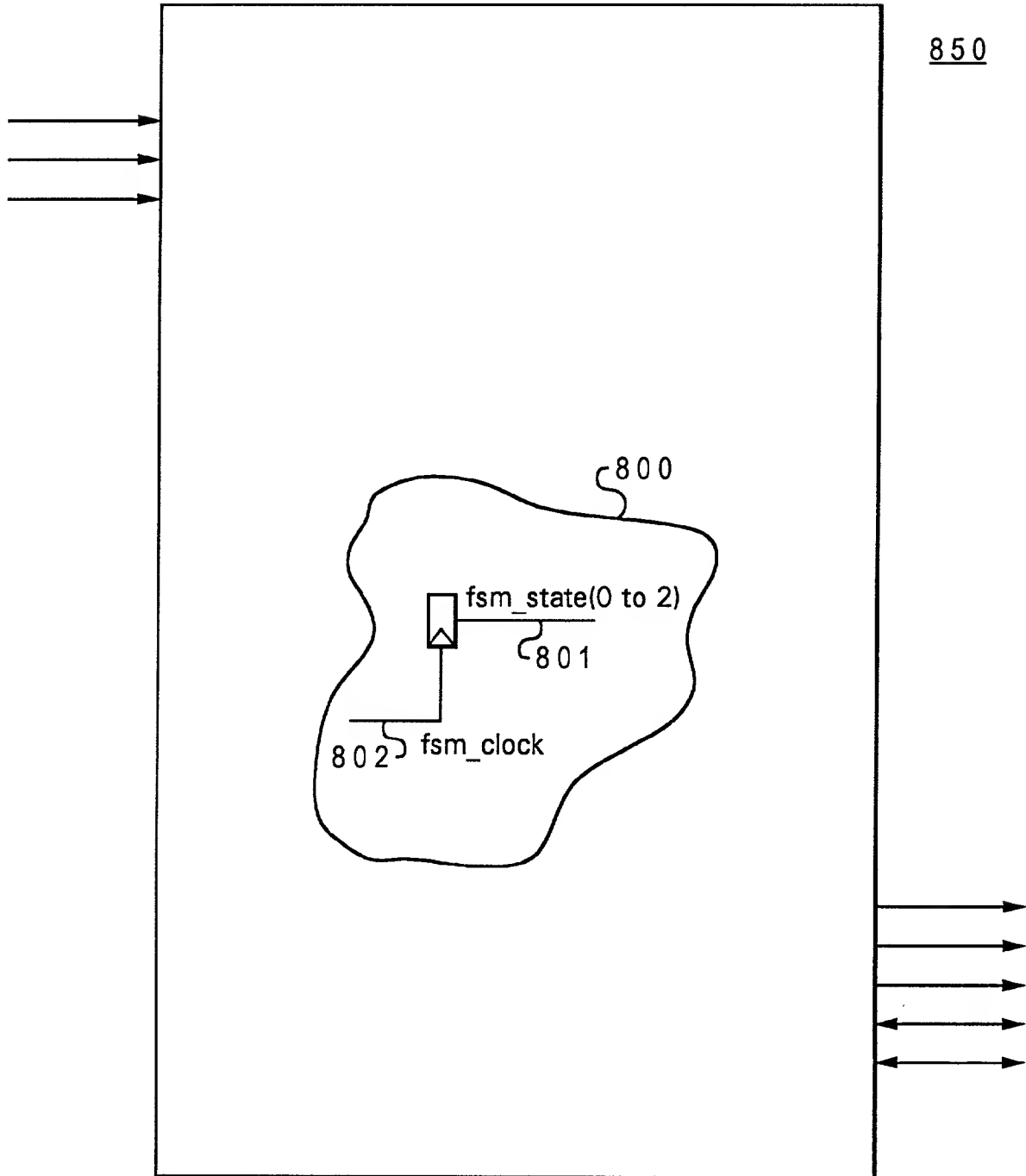


Fig. 8B
Prior Art

19/62

ENTITY FSM IS

```
PORT(
    ....ports for entity fsm....
);
```

ARCHITECTURE FSM OF FSM IS

BEGIN

... HDL code for FSM and rest of the entity ...

fsm_state(0 to 2) <= ... Signal 801 ...

```

8 5 3 { --!! Embedded FSM : examplefsm;
8 5 9 { --!! clock      : (fsm_clock);
8 5 4 { --!! state_vector : (fsm_state(0 to 2));
8 5 5 { --!! states      : (S0, S1, S2, S3, S4);
8 5 6 { --!! state_encoding : ('000', '001', '010', '011', '100');
      { --!! arcs        : (S0 => S0, S0 => S1, S0 => S2,
8 5 7 { --!!              (S1 => S2, S1 => S3, S2 => S2,
      { --!!              (S2 => S3, S3 => S4, S4 => S0);
8 5 8 { --!! End FSM;

```

END;

Fig. 8C

20/62

entity FSM : FSM

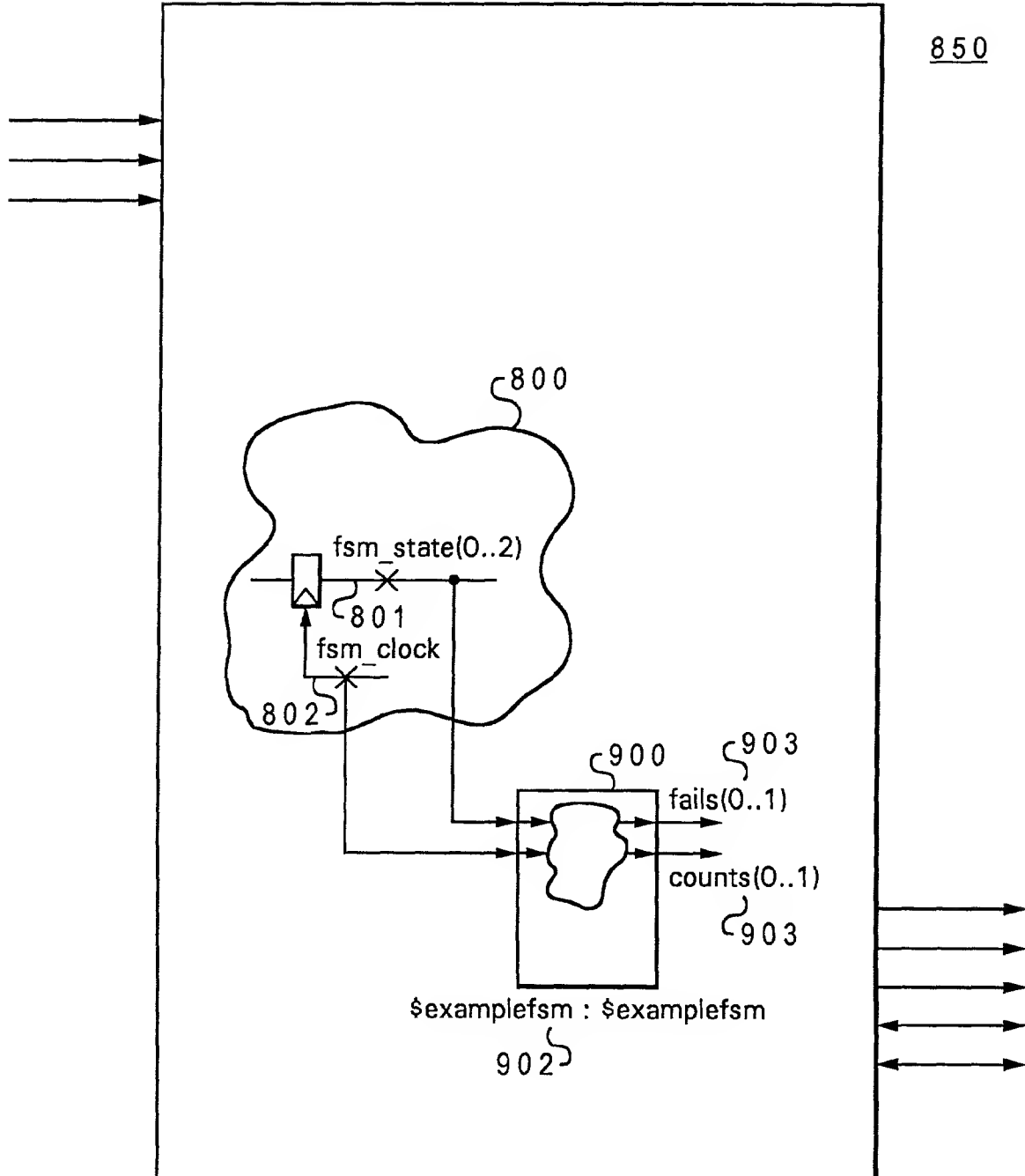
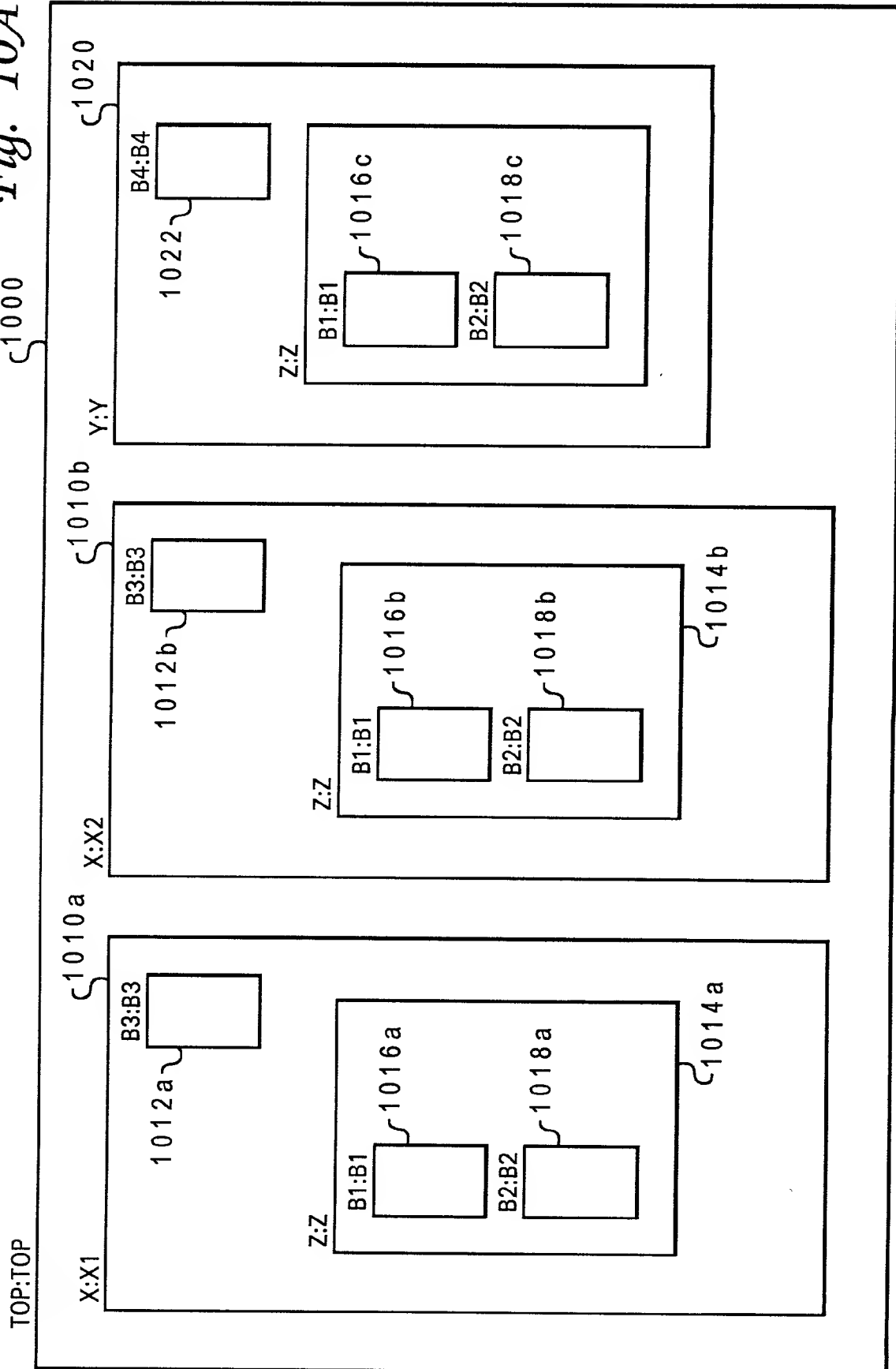


Fig. 9

Fig. 10A



22/62

1030 1032 1034 1036
<instantiation identifier> . <instrumentation entity name> . <design entity name> . <eventname>

Fig. 10B

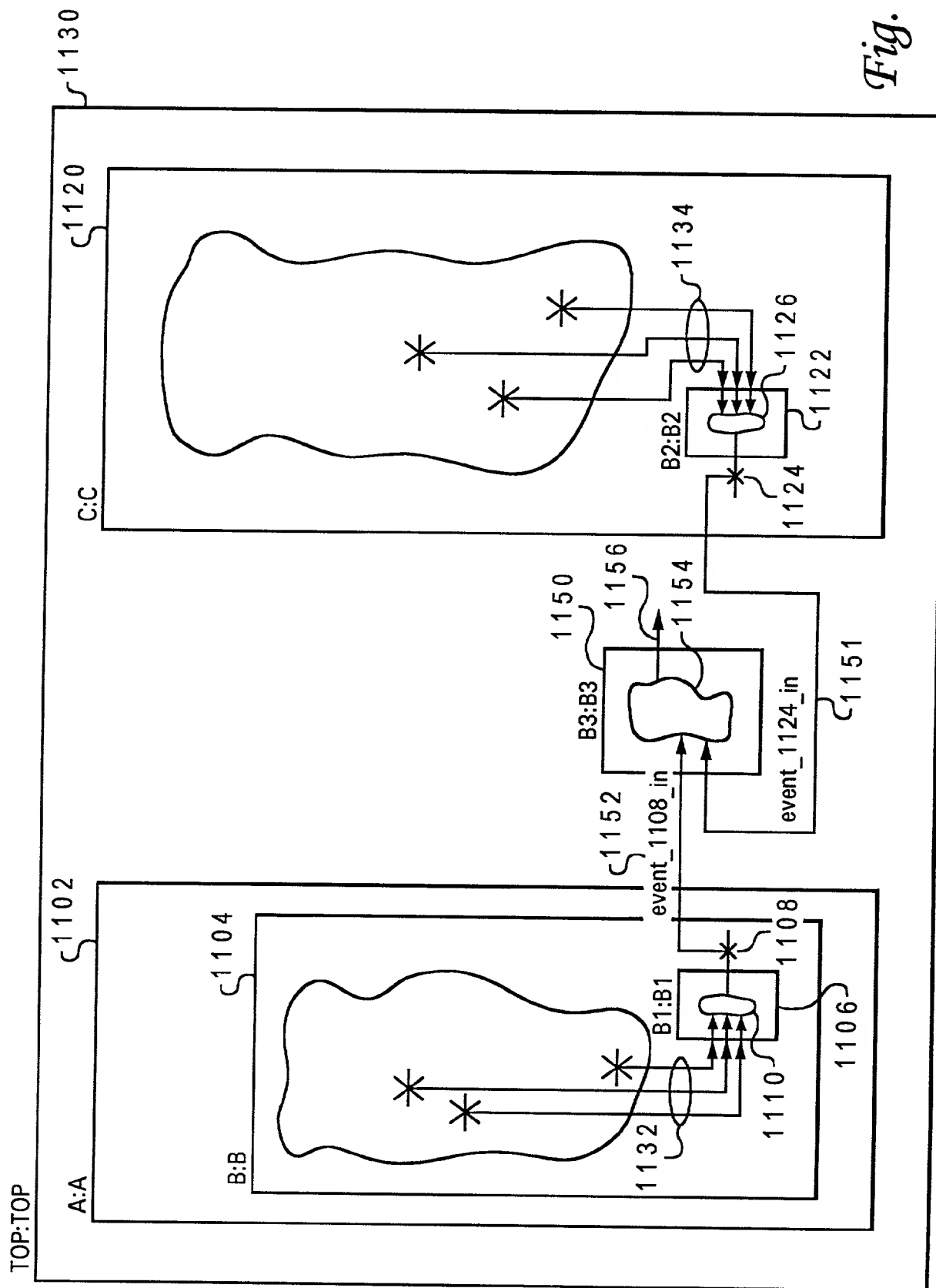
1030 1032 1034 1036
X1 B3 X COUNT1 1040
X1.Z B1 Z COUNT1 1041
X1.Z B2 Z COUNT1 1042
X2 B3 X COUNT1 1043
X2.Z B1 Z COUNT1 1044
X2.Z B2 Z COUNT1 1045
Y B4 Y COUNT1 1046
Y.Z B1 Z COUNT1 1047
Y.Z B2 Z COUNT1 1048

Fig. 10C

1030 1034 1036
<instantiation identifier> . <design entity name> . <eventname>

Fig. 10D

Fig. 11A



24/62

--!! Inputs
--!! event_1108_in <= C.[B2.count.event_1108];
--!! event_1124_in <= A.B.[B1.count.event_1124];
--!! End Inputs

Fig. 11B

--!! Inputs
--!! event_1108_in <= C.[count.event_1108];
--!! event_1124_in <= B.[count.event_1124];
--!! End Inputs

Fig. 11C

20120220-20120220

25/62

Fig. 12A

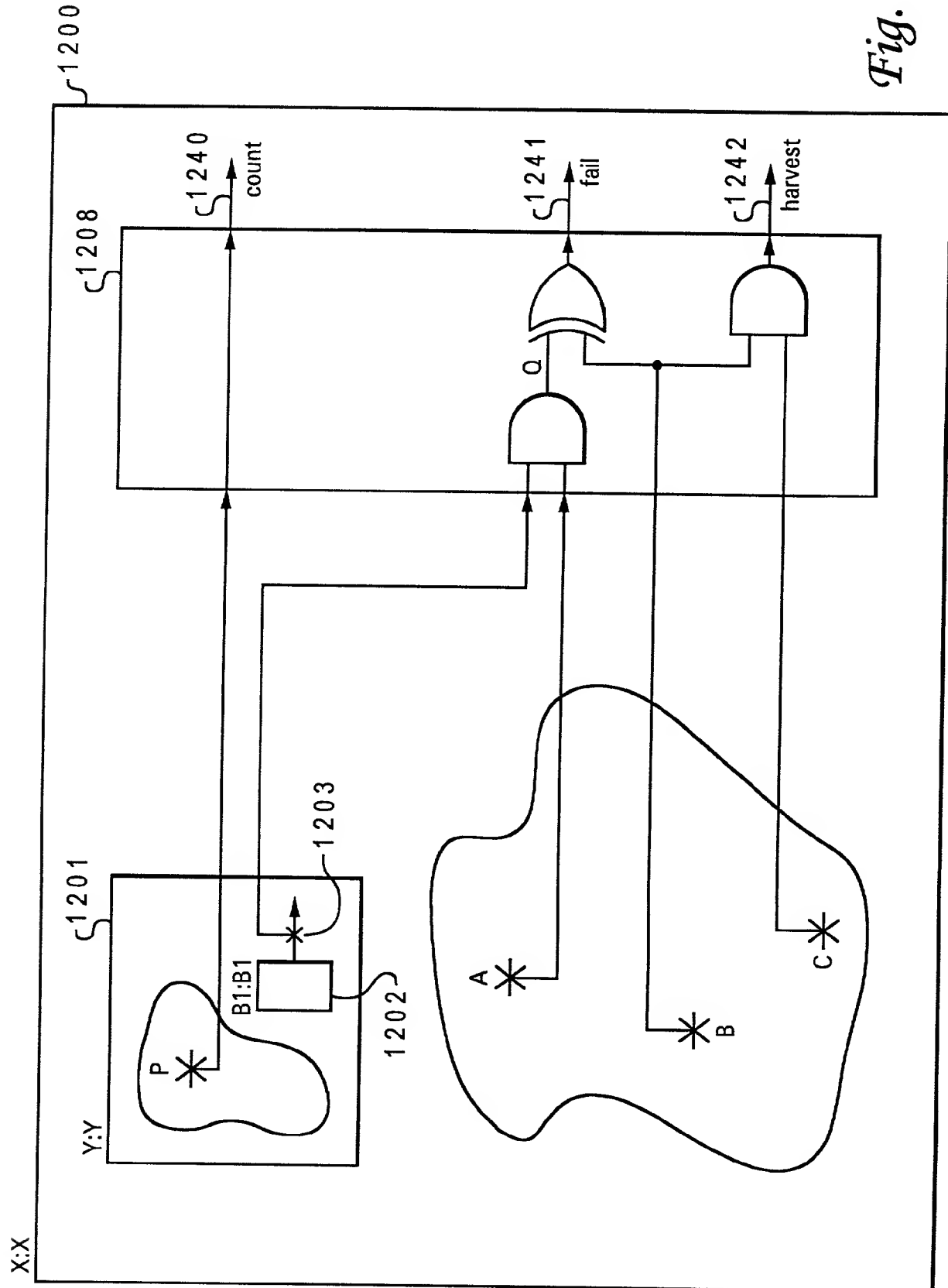


Figure 1. The structure of the proposed model. The model consists of three main parts: a feature extraction module, a classification module, and a fusion module. The feature extraction module uses a 3D CNN to extract features from the input data. The classification module uses a 2D CNN to extract features from the input data. The fusion module combines the features from the 3D and 2D CNNs to produce the final output.

Figure 1. The structure of the proposed model. The model consists of three main parts: a feature extraction module, a classification module, and a fusion module. The feature extraction module uses a 3D CNN to extract features from the input data. The classification module uses a 2D CNN to extract features from the input data. The fusion module combines the features from the 3D and 2D CNNs to produce the final output.

27/62

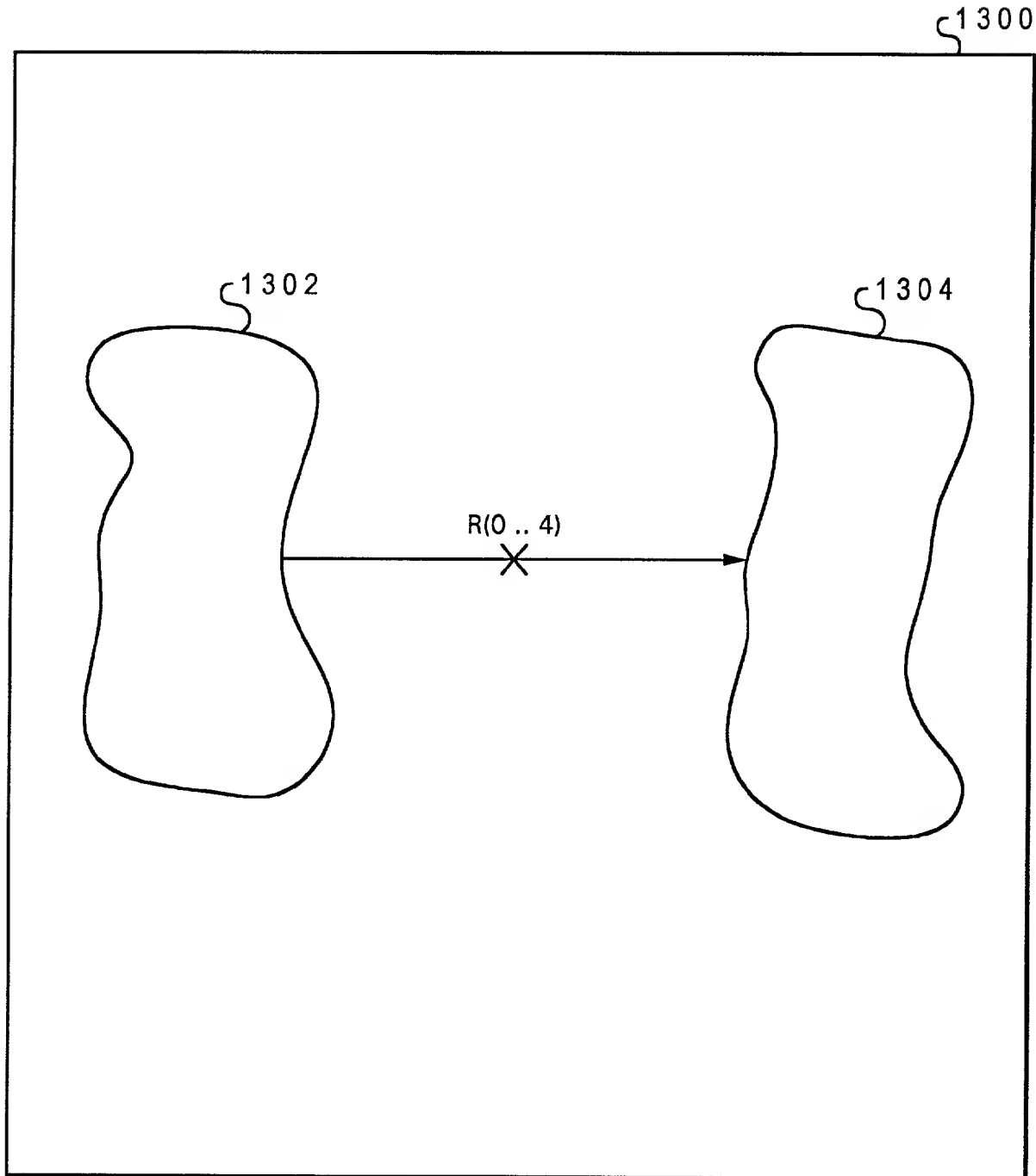
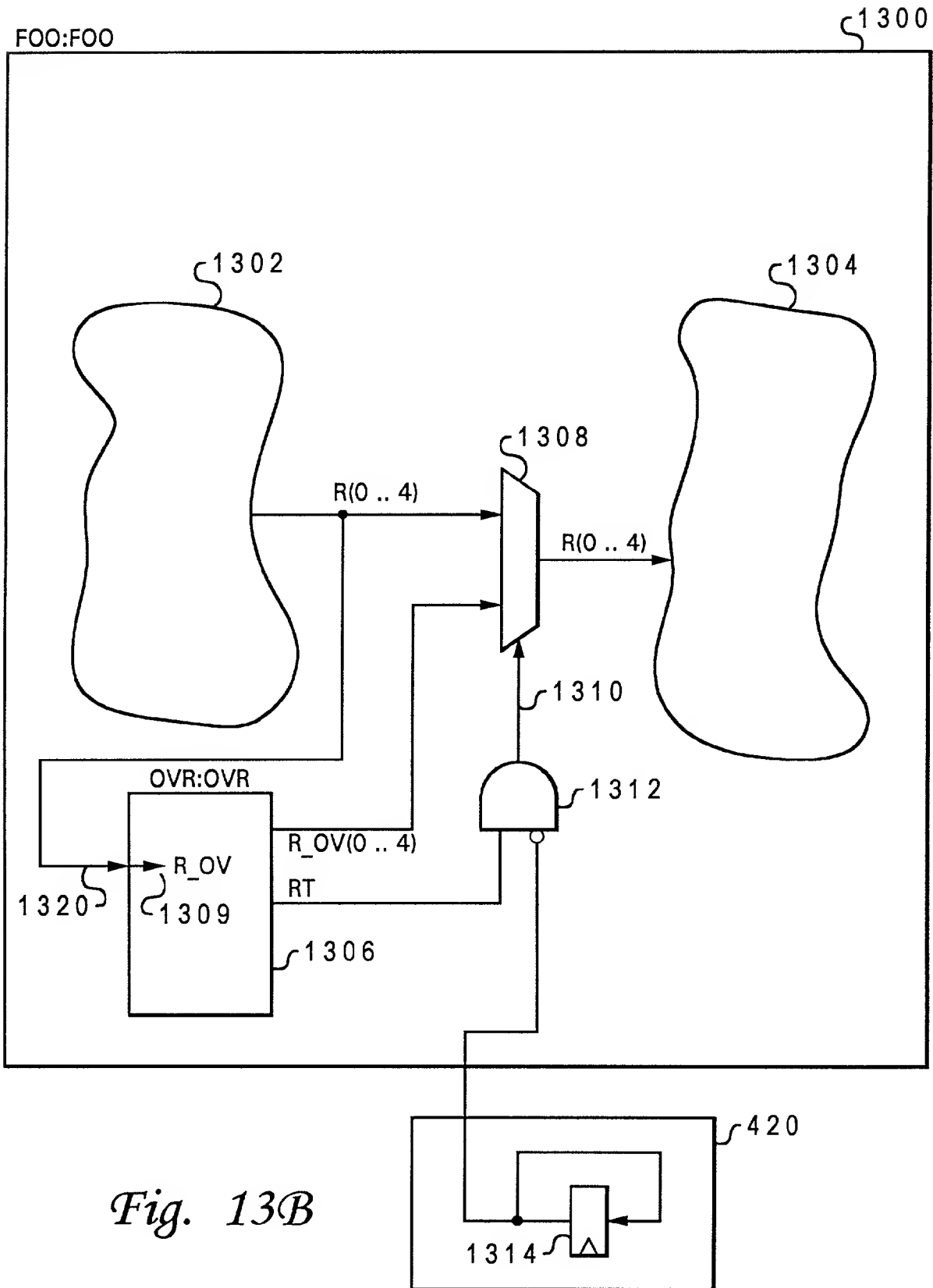


Fig. 13A

28/62



29/62

```

ENTITY OVR IS
    PORT(  R_IN      :  IN std_ulogic_vector(0 .. 4);
          .
          .
          ... other ports as required ...
          .
          .
          R_OV      :  OUT std_ulogic_vector(0 .. 4);
          RT        :  OUT std_ulogic
    );

--!! BEGIN
--!! Design Entity: FOO;

--!! Inputs (0 to 4)
--!! R_IN => {R(0 .. 4)};
--!! :
... other ports as needed ...
--!! :
--!! End Inputs

--!! Outputs
--!! <R_OVRIDE> : R_OV(0 .. 4) => R(0 .. 4) [RT];
--!! End Outputs

--!! End

ARCHITECTURE example of OVR IS
BEGIN
    ... HDL code for entity body section ...
END;

```

1364

1362

1363

1360

1361

1356

1351

1340

1358

Fig. 13C

30/62

ENTITY FOO IS

PORT(:
:
:;
);

ARCHITECTURE example of FOO IS

BEGIN

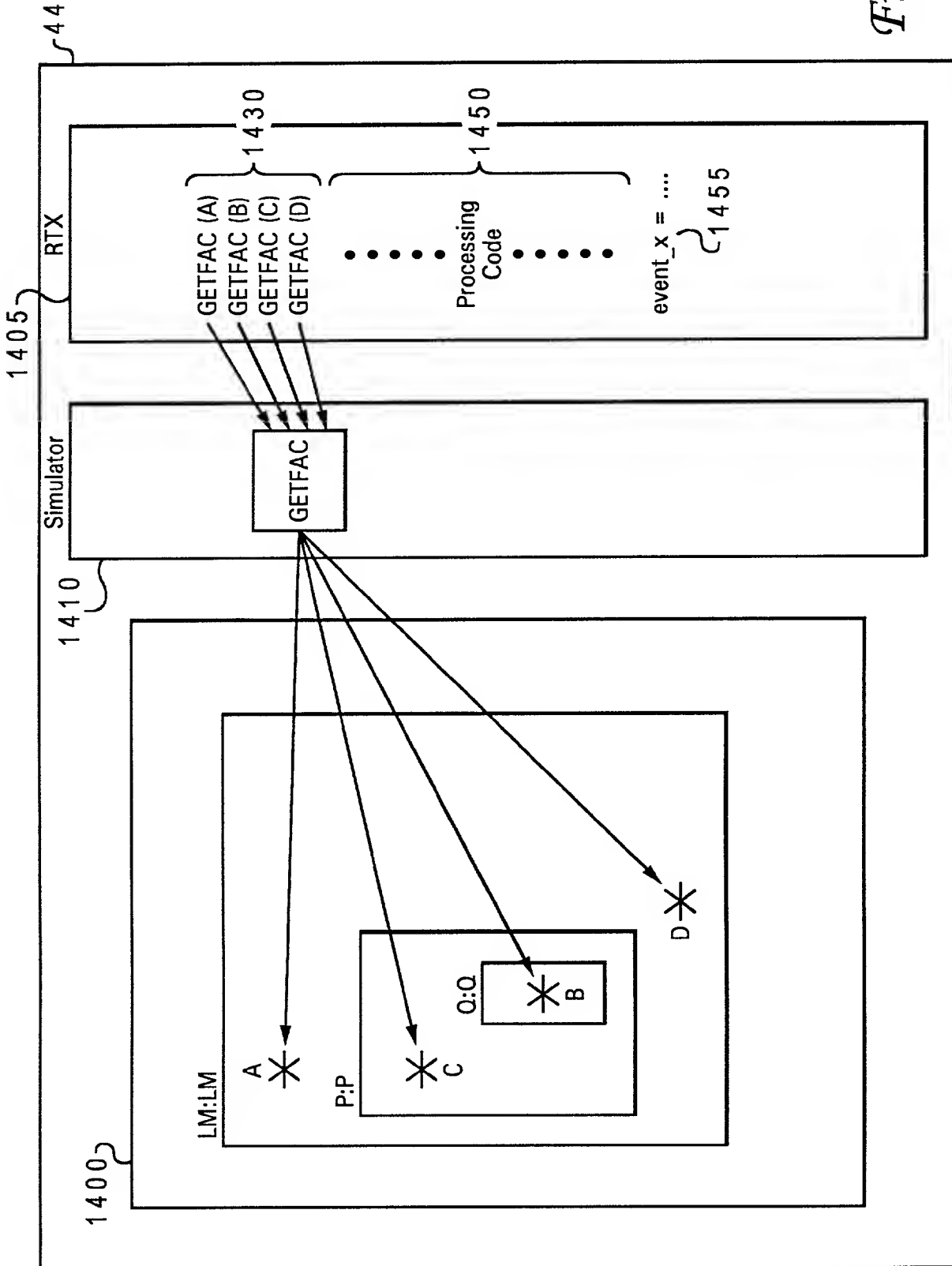
.
.
.
.
.
R <=
.
.
.
.

1380 { --!! R_IN <= {R}; 1381
--!! 1382
--!!
--!! R_OV(0 to 4) <=; 1383
--!! RT <=;
--!! [override, R_OVRRIDE, R(0 .. 4), RT] <= R_OV(0 to 4);
1384

Fig. 13D

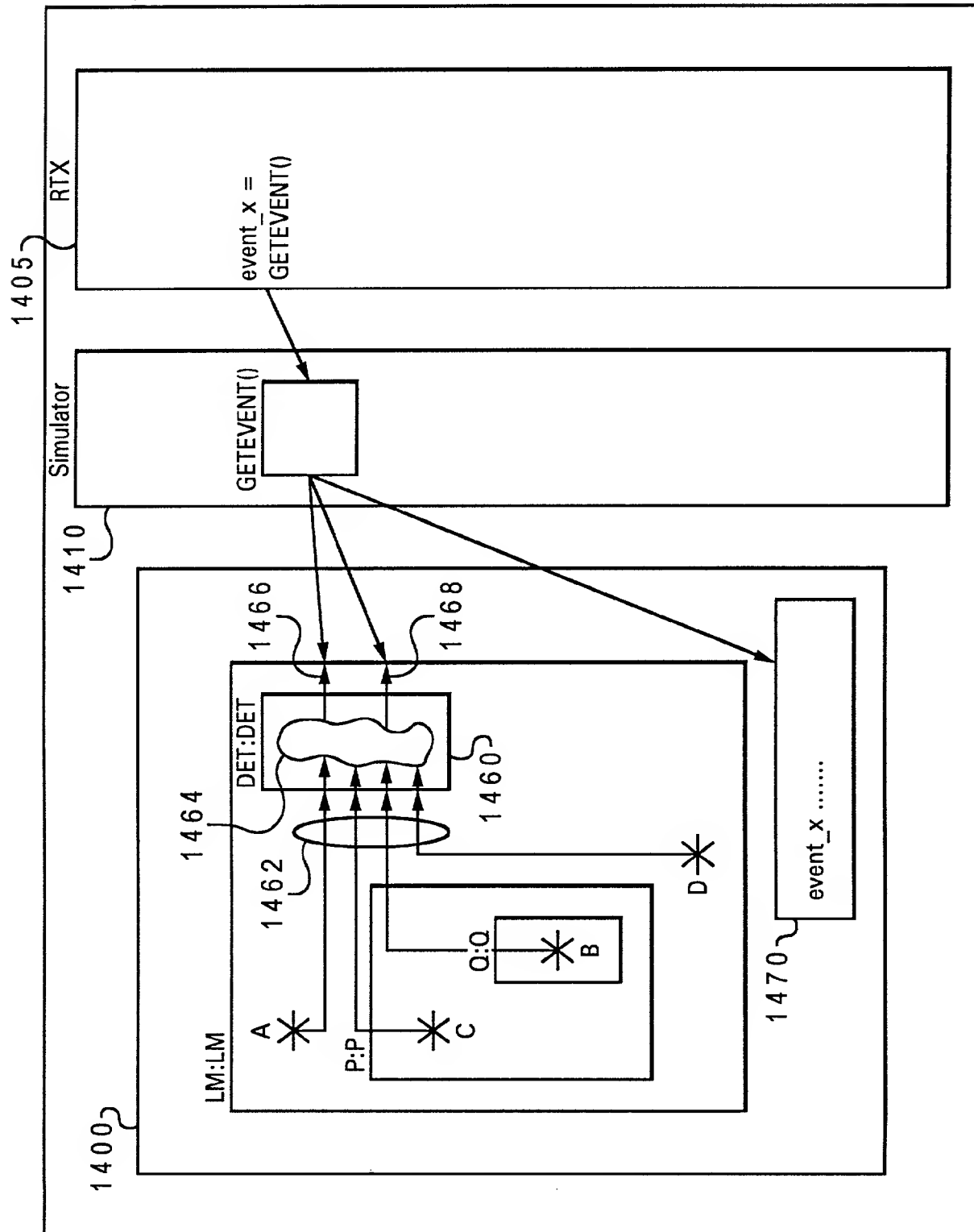
31/62

Fig. 14A



32/62

Fig. 14B



33/62

```

ENTITY DET IS
    PORT(
        A      : IN std_ulogic;
        B      : IN std_ulogic_vector(0 to 5);
        C      : IN std_ulogic;
        D      : IN std_ulogic;
        :      :
        event_x : OUT std_ulogic_vector(0 to 2);
        x_here  : OUT std_ulogic;
    );

    --!! BEGIN
    --!! Design Entity: LM;

    --!! Inputs
    --!! A  => A;
    --!! B  => P.Q.B;
    --!! C  => P.C;
    --!! D  => D;
    --!! End Inputs

    --!! Detections
    --!! <event_x>:event_x(0 to 2) [x_here];
    --!! End Detections

    --!! End;

    ARCHITECTURE example of DET IS
    BEGIN
        ... HDL code ...

    END;

```

1491 {

1493 {

1495 {

1494 {

1480 {

Fig. 14C

34/62

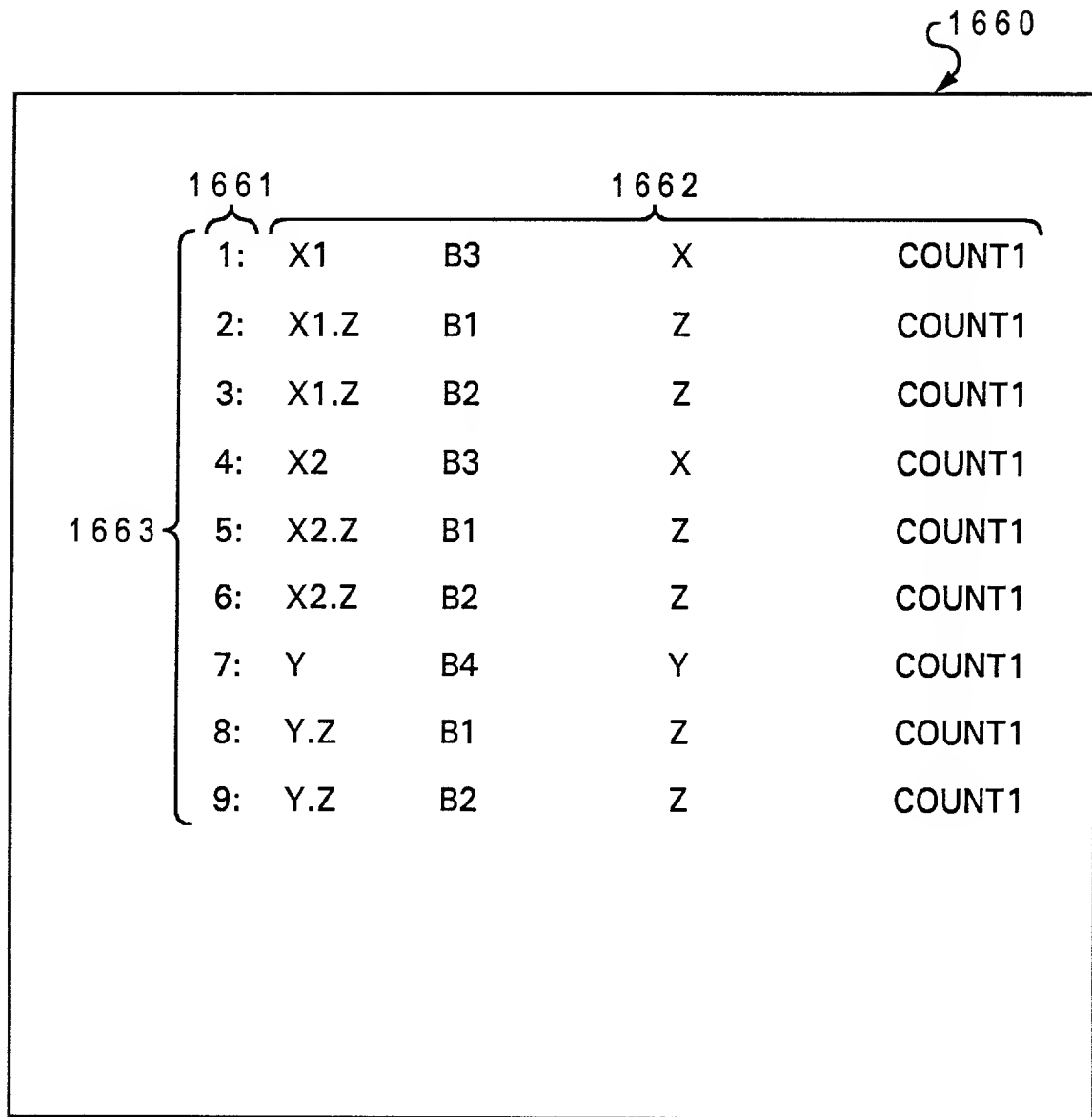
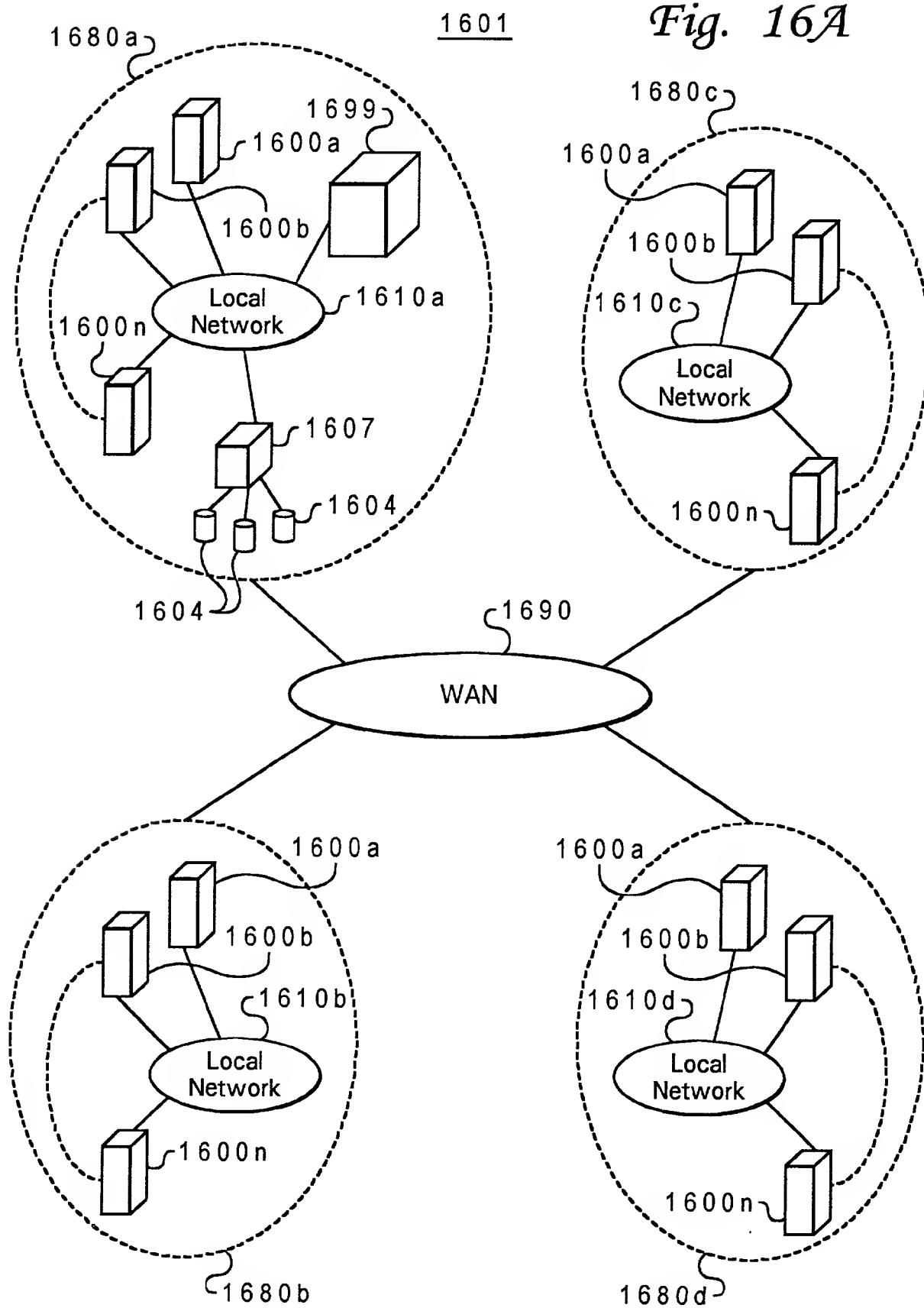


Fig. 15

35/62

Fig. 16A



36/62

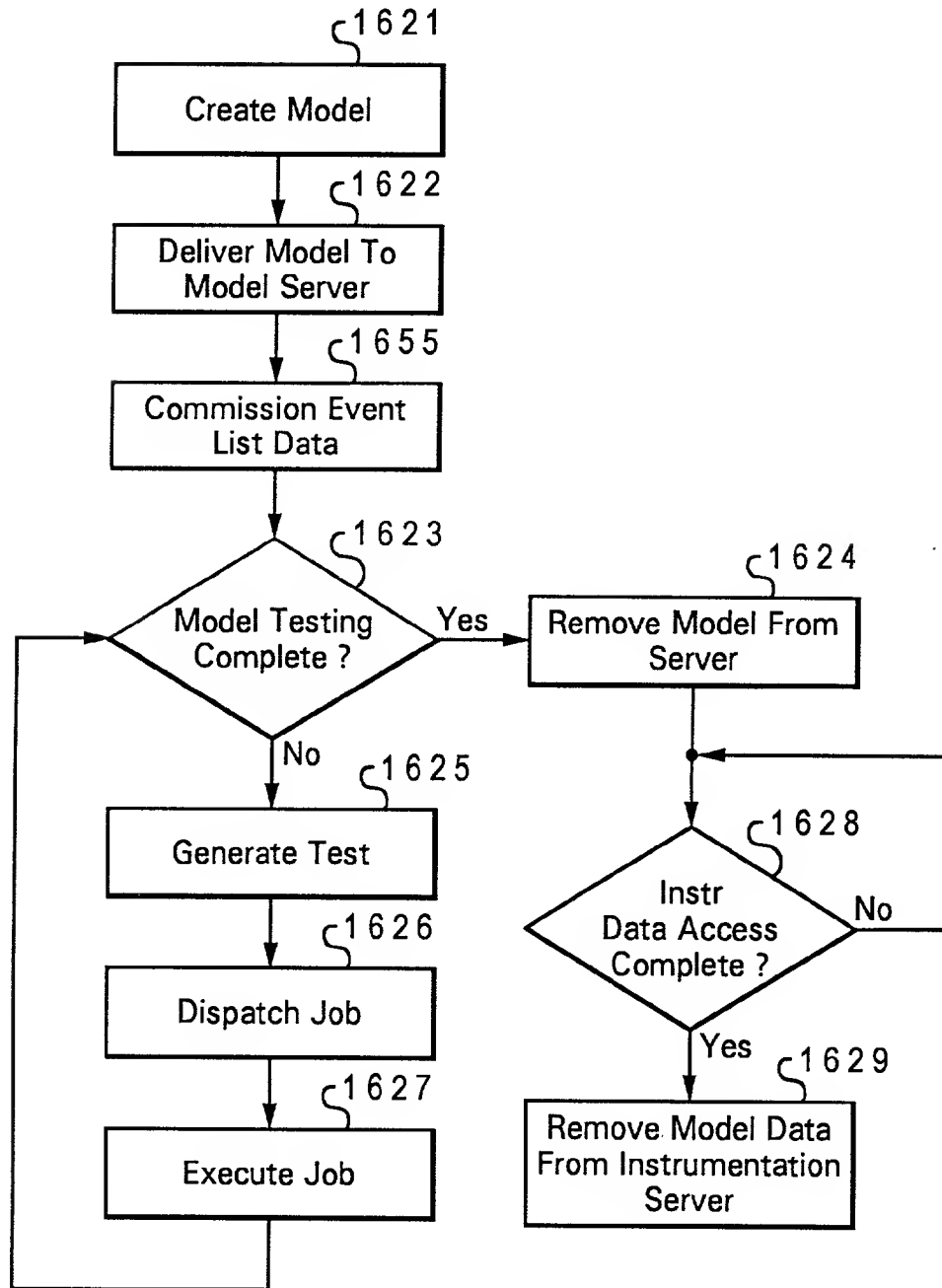


Fig. 16B

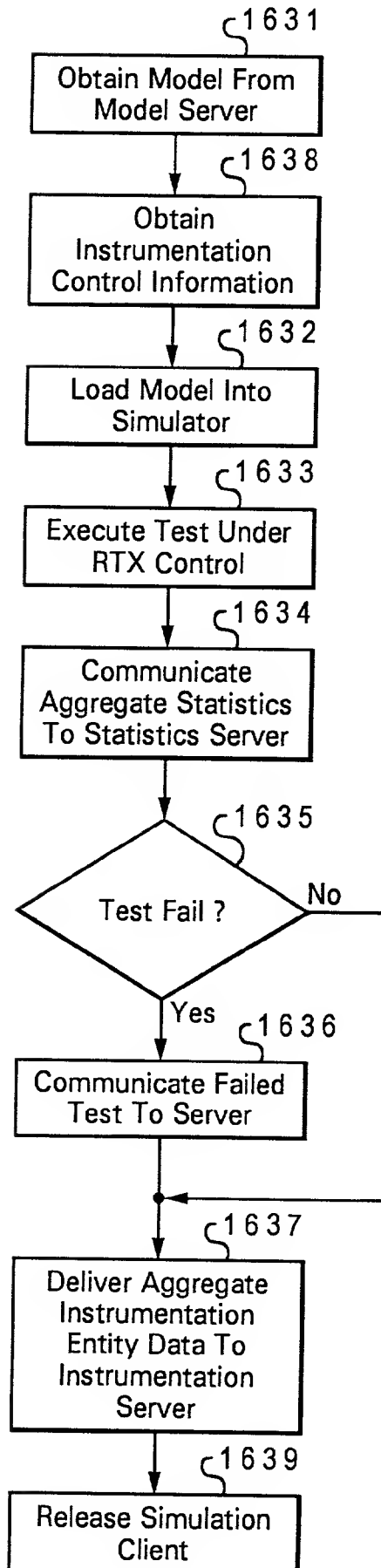


Fig. 16C

38/62

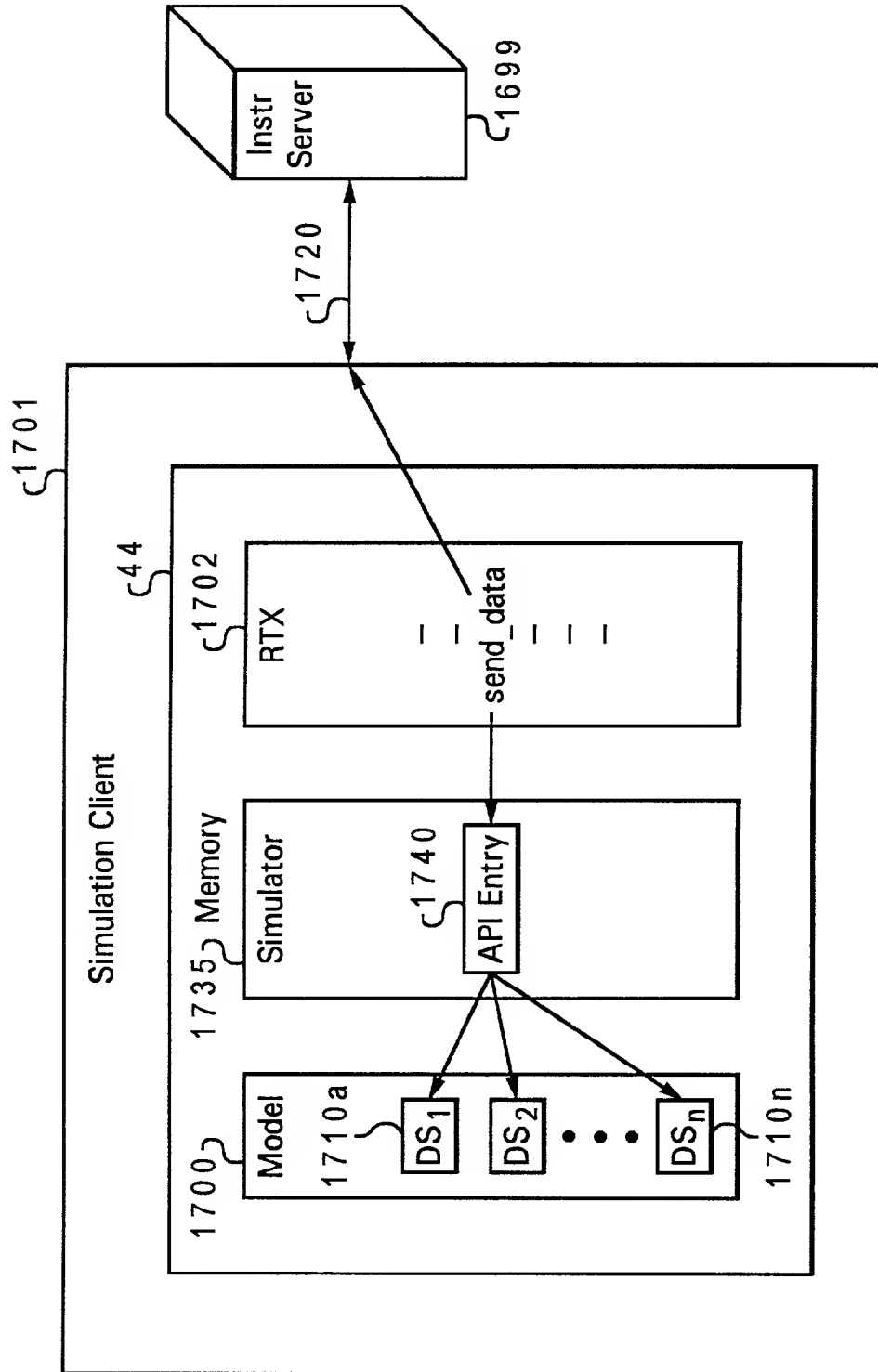


Fig. 17A

39/62

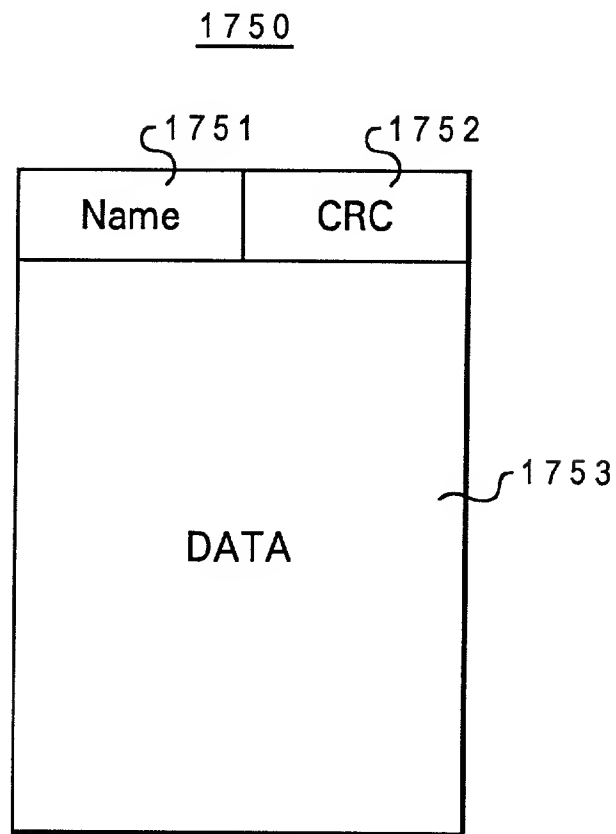


Fig. 17B

40/62

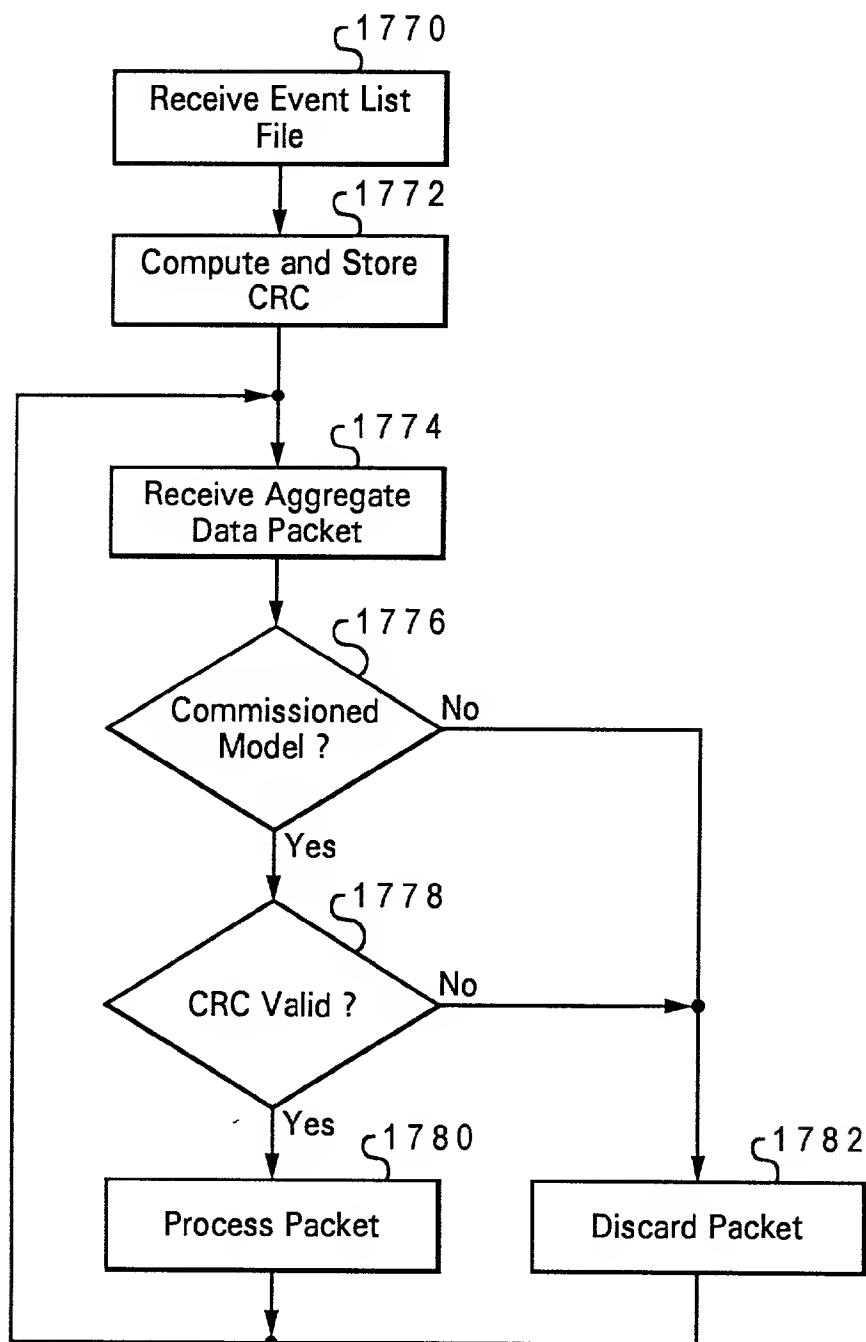


Fig. 17C

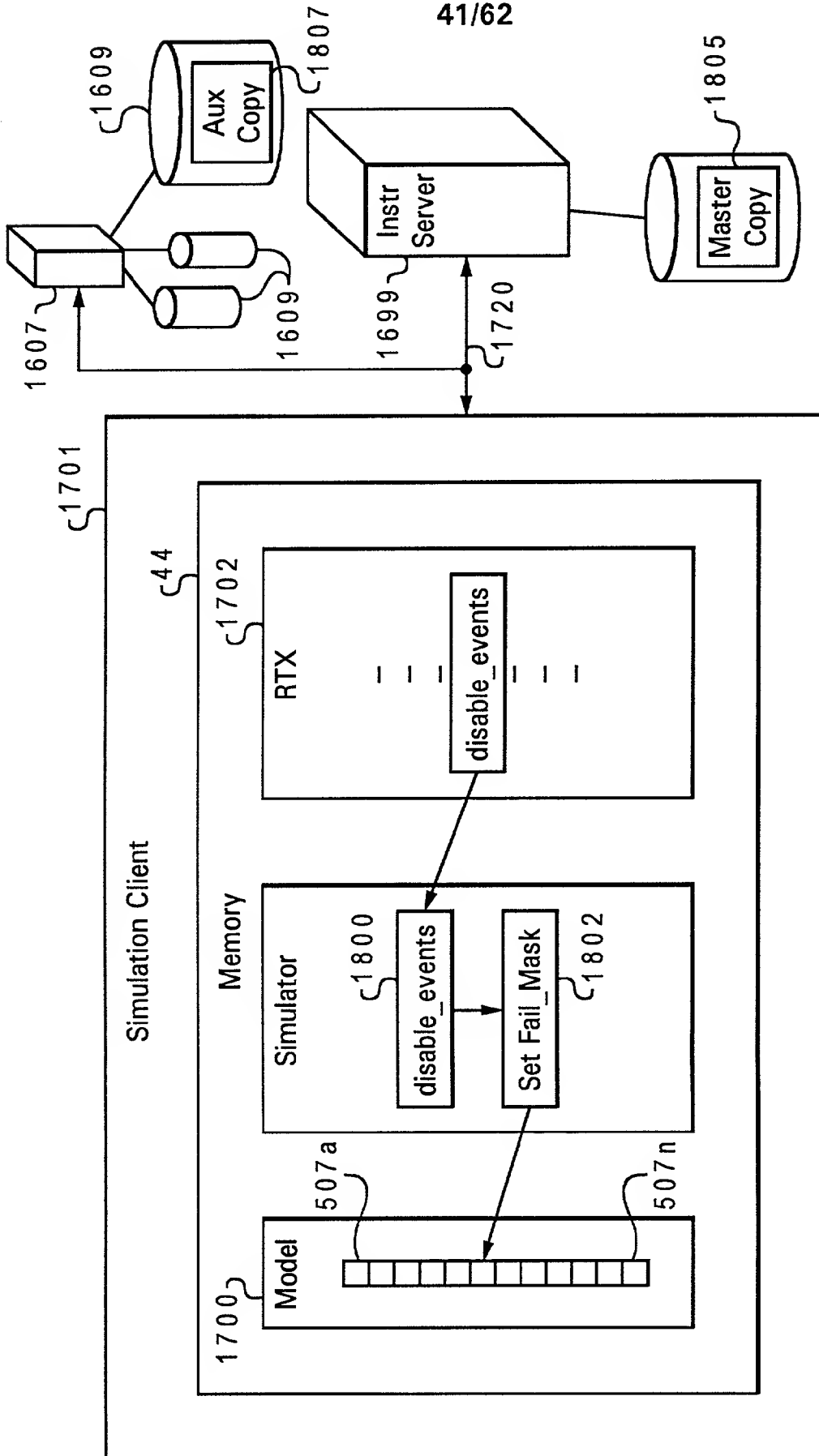
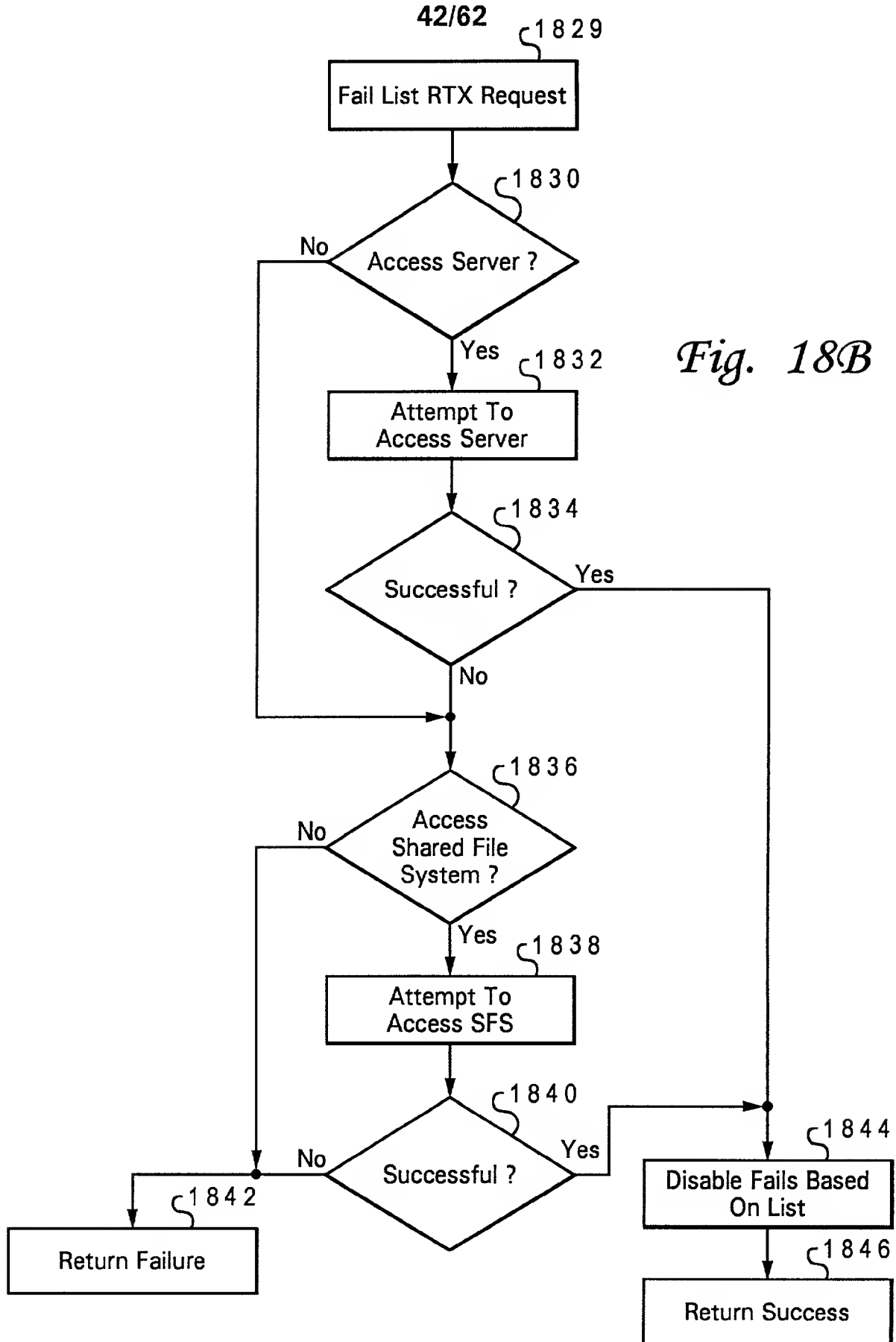


Fig. 18A

42/62



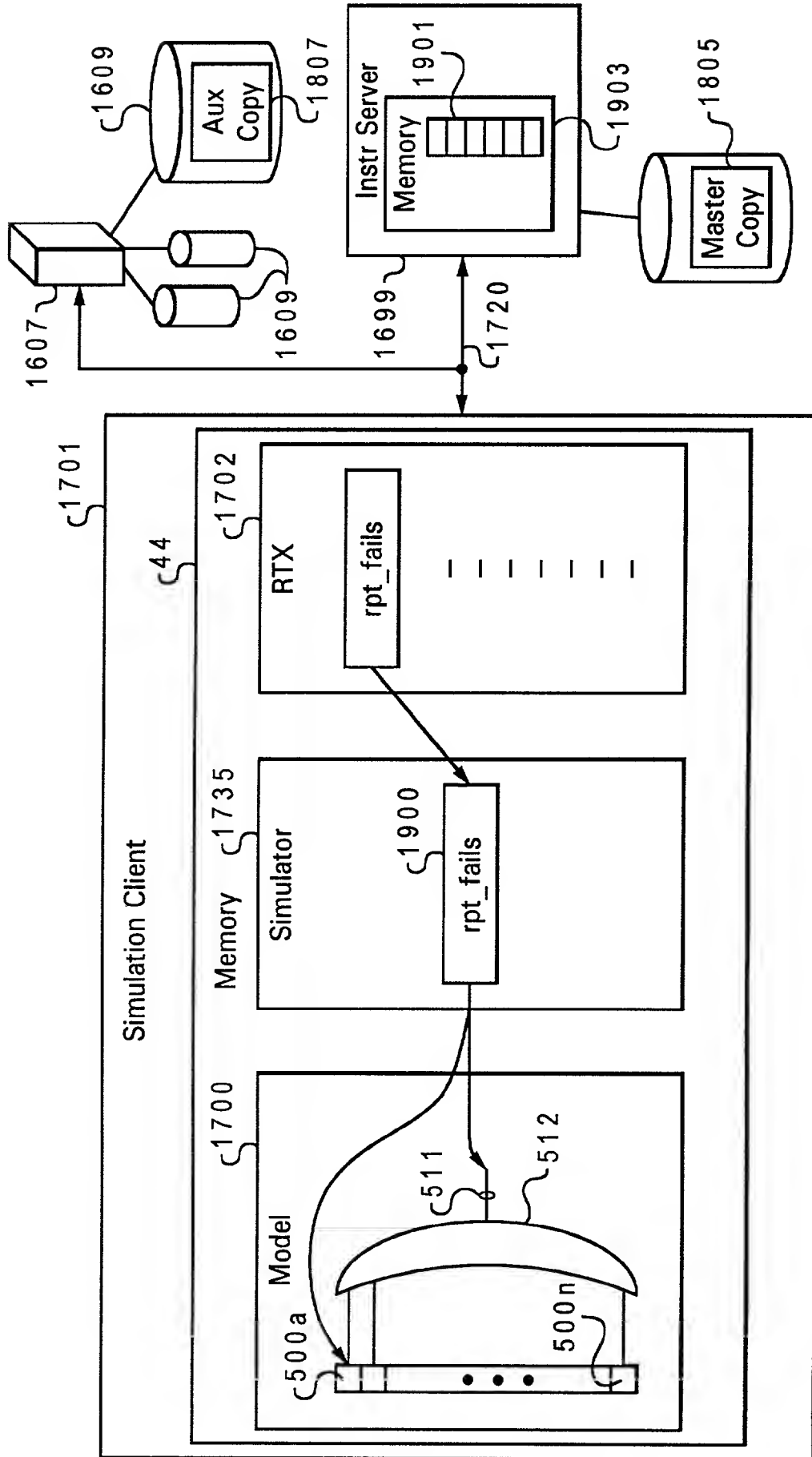
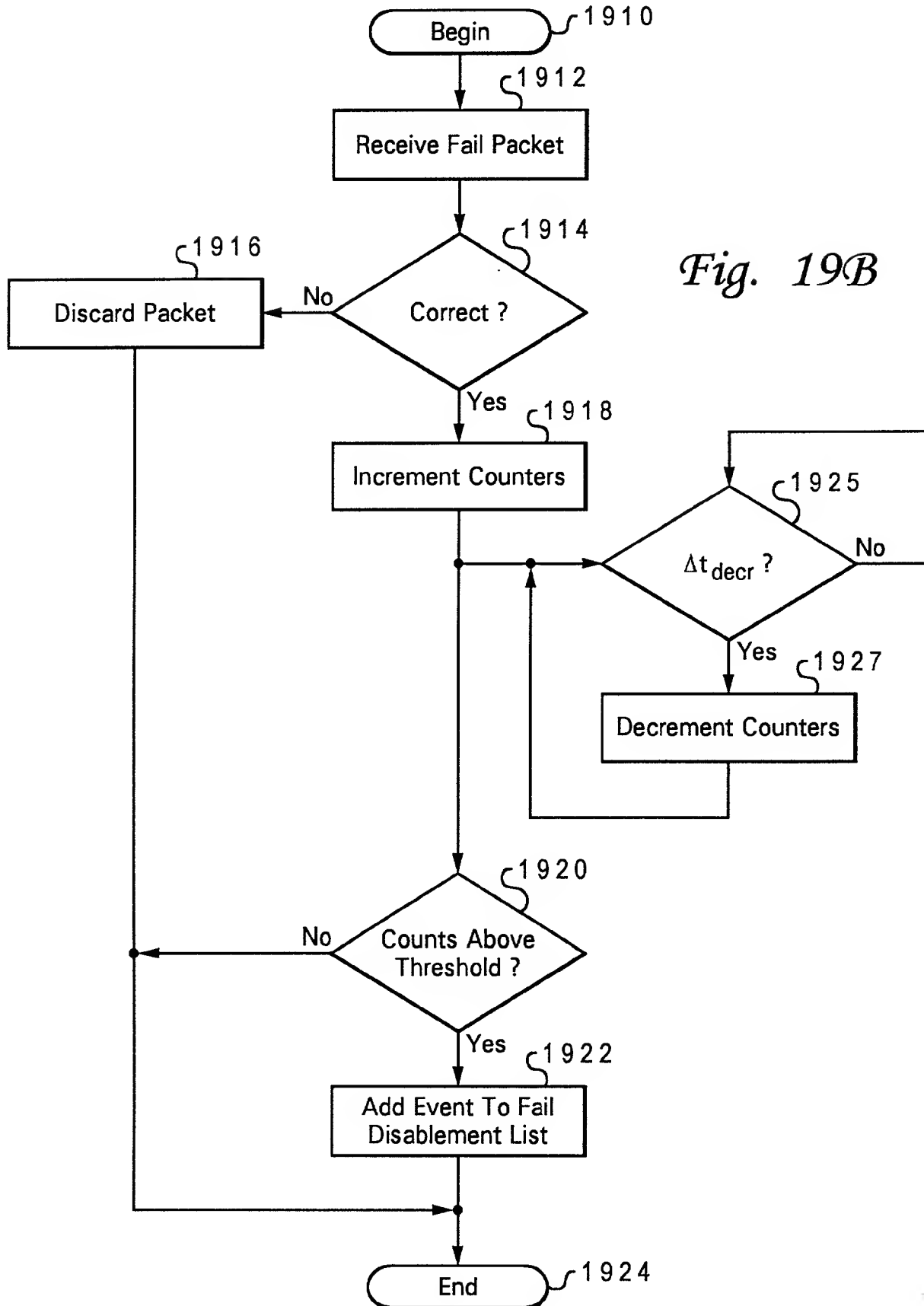


Fig. 19A

44/62



45/62

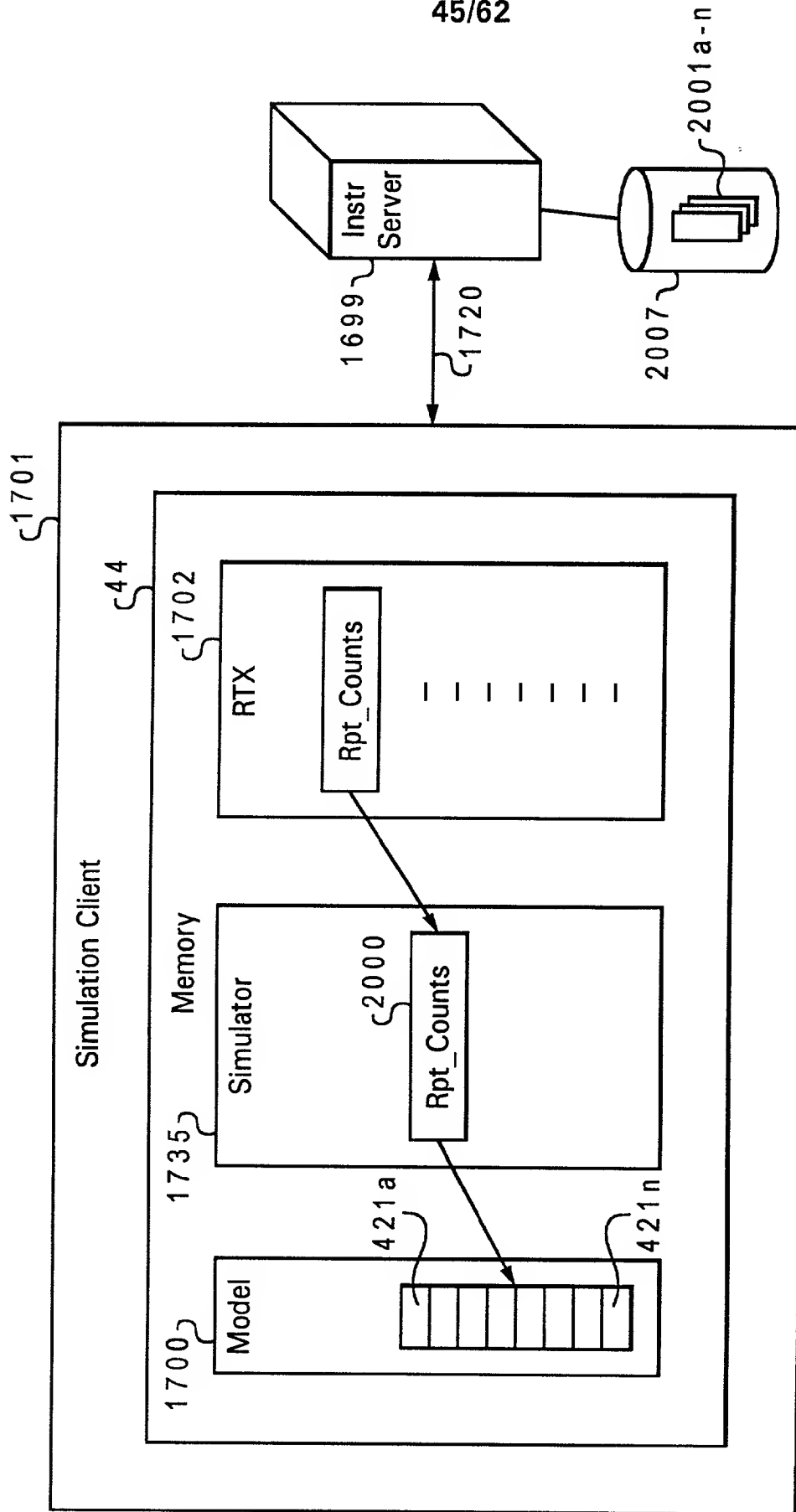


Fig. 20A

46/62

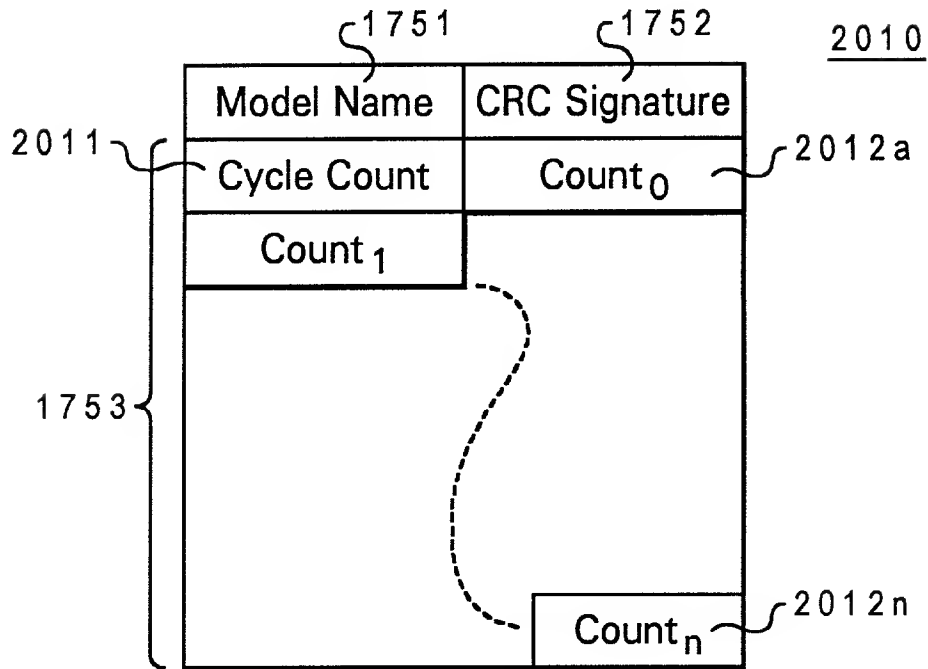


Fig. 20B

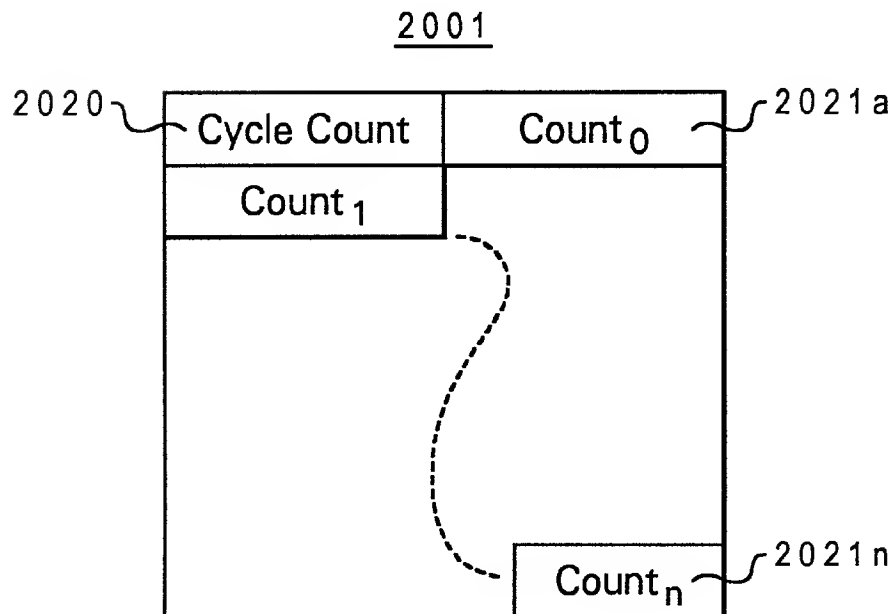


Fig. 20C

47/62

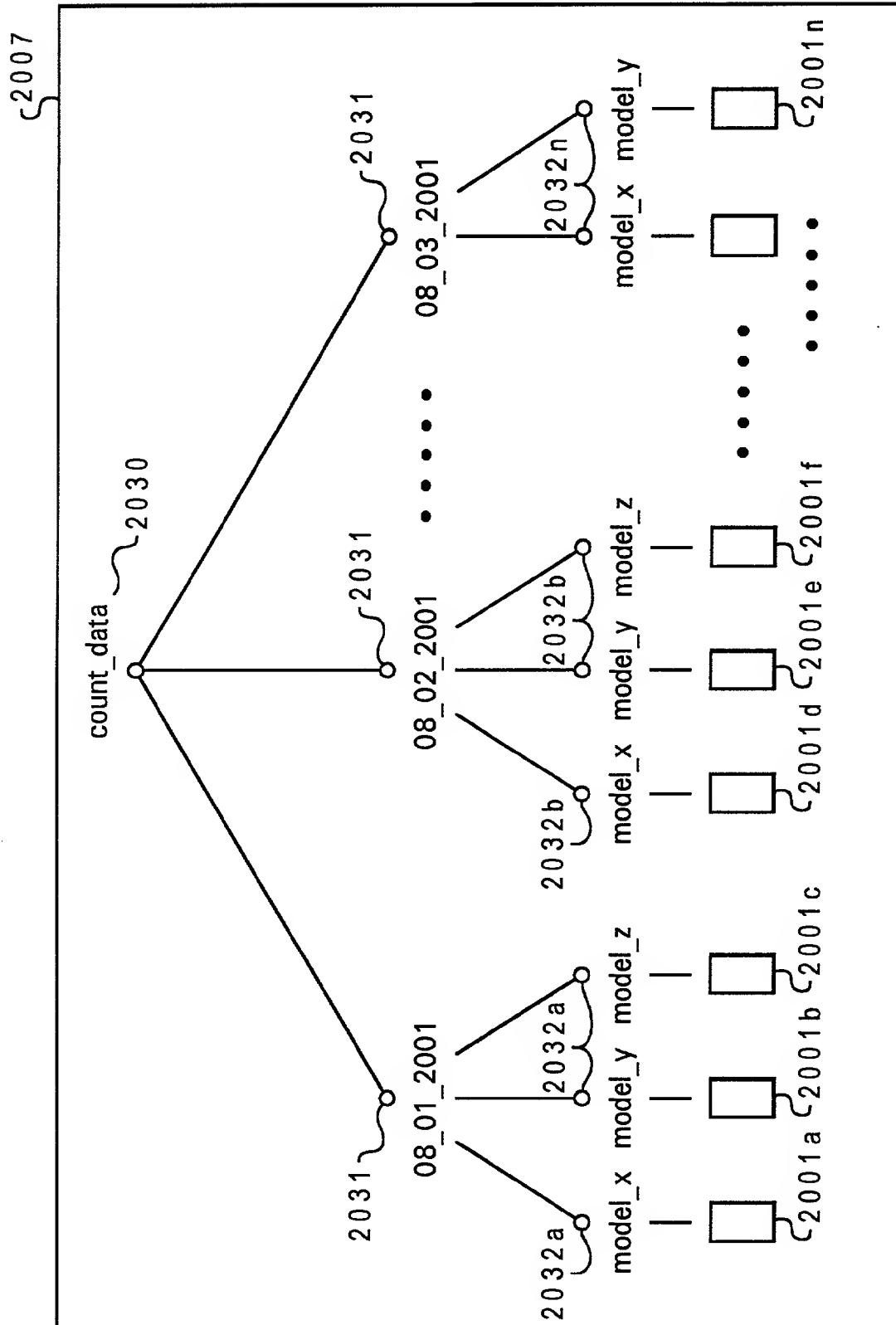


Fig. 20D

48/62

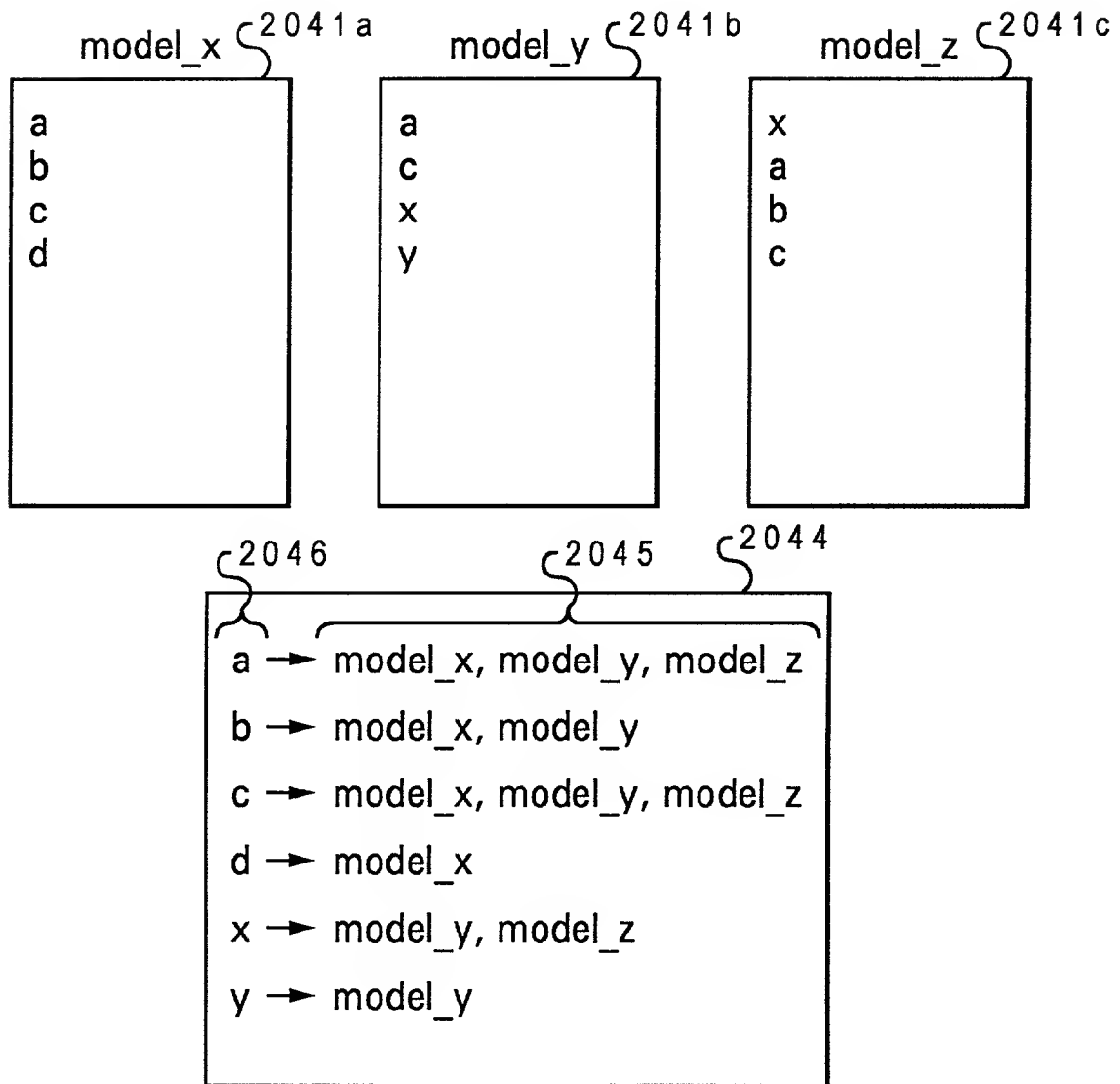
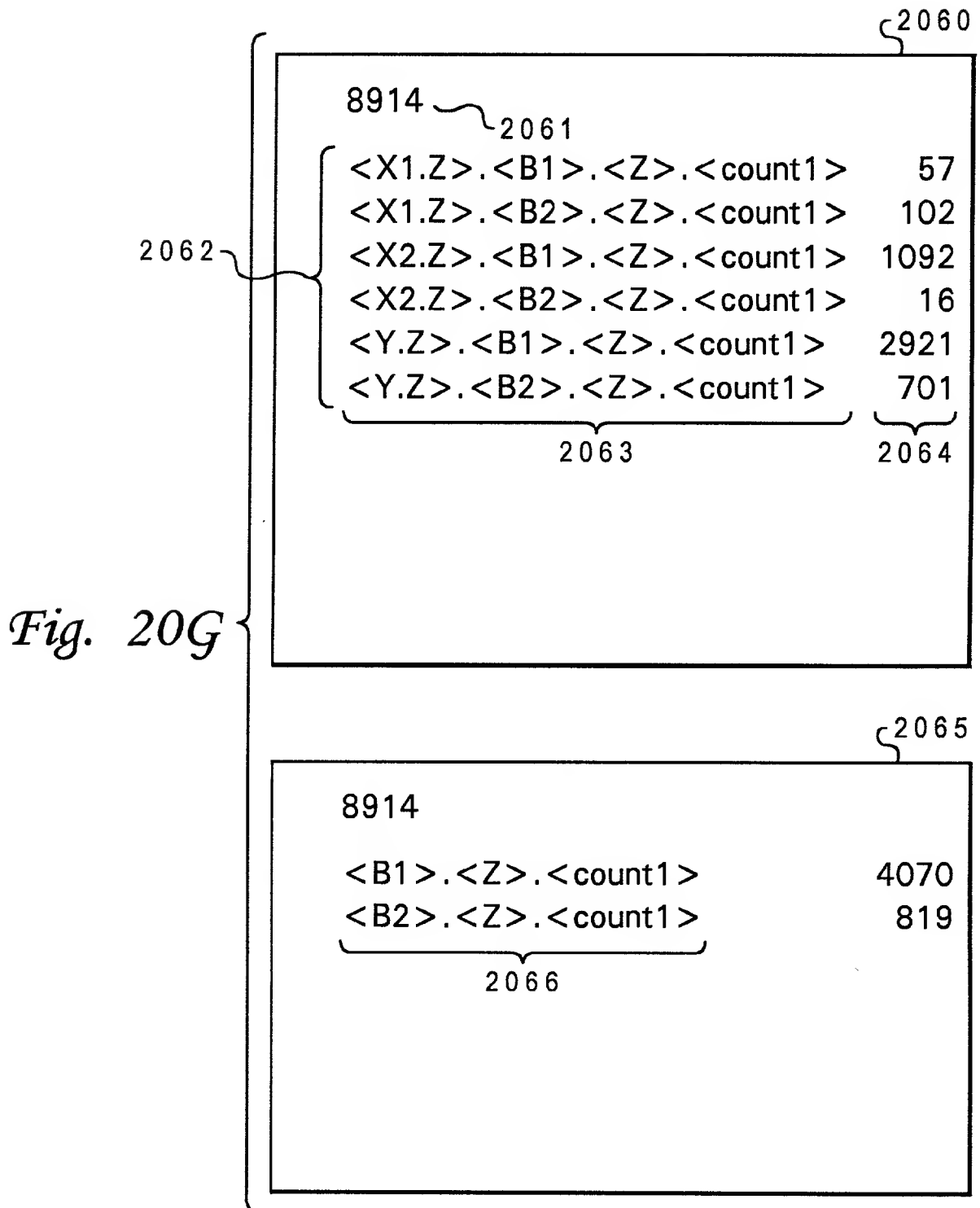


Fig. 20E

50/62



51/62

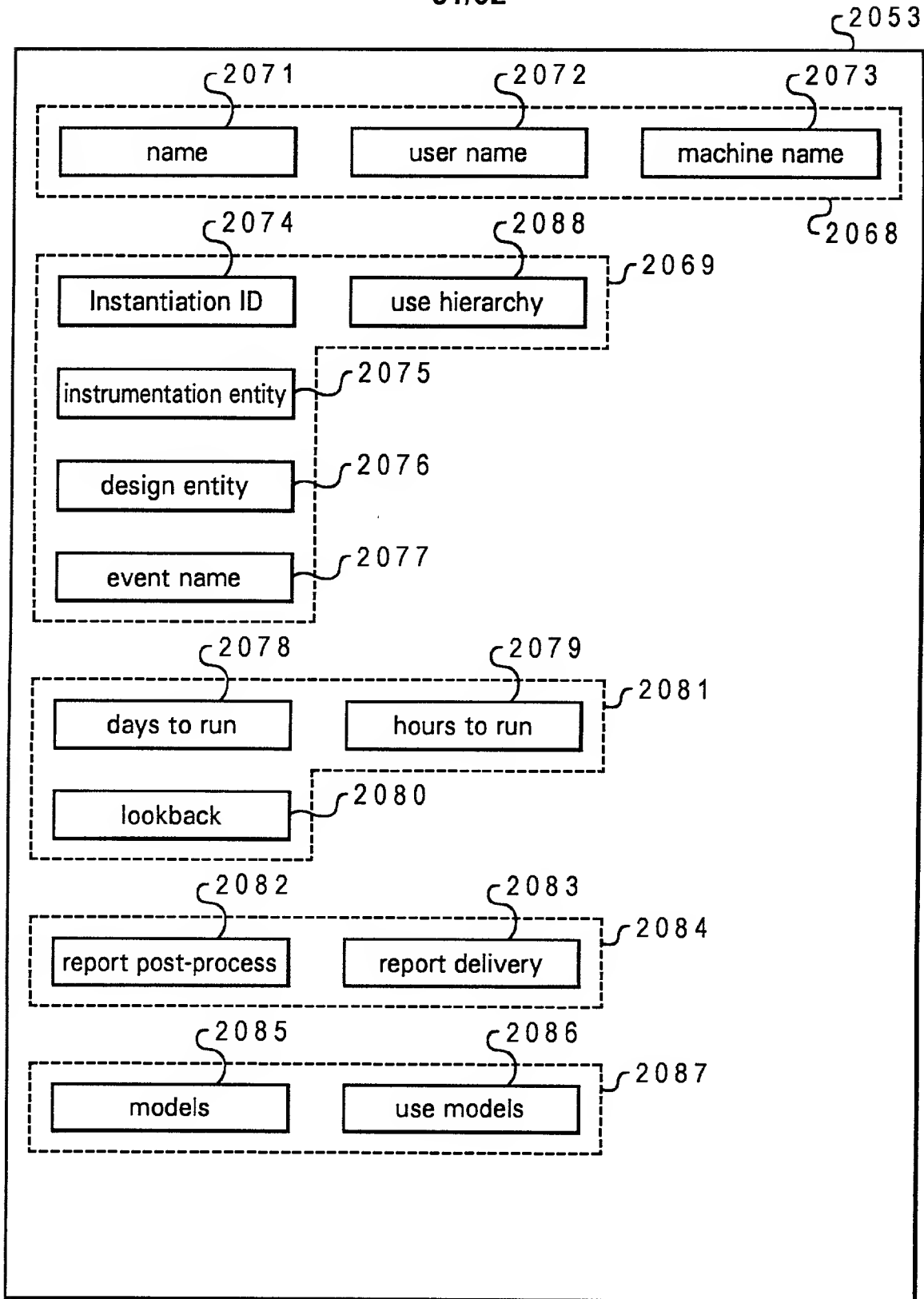


Fig. 20H

52/62

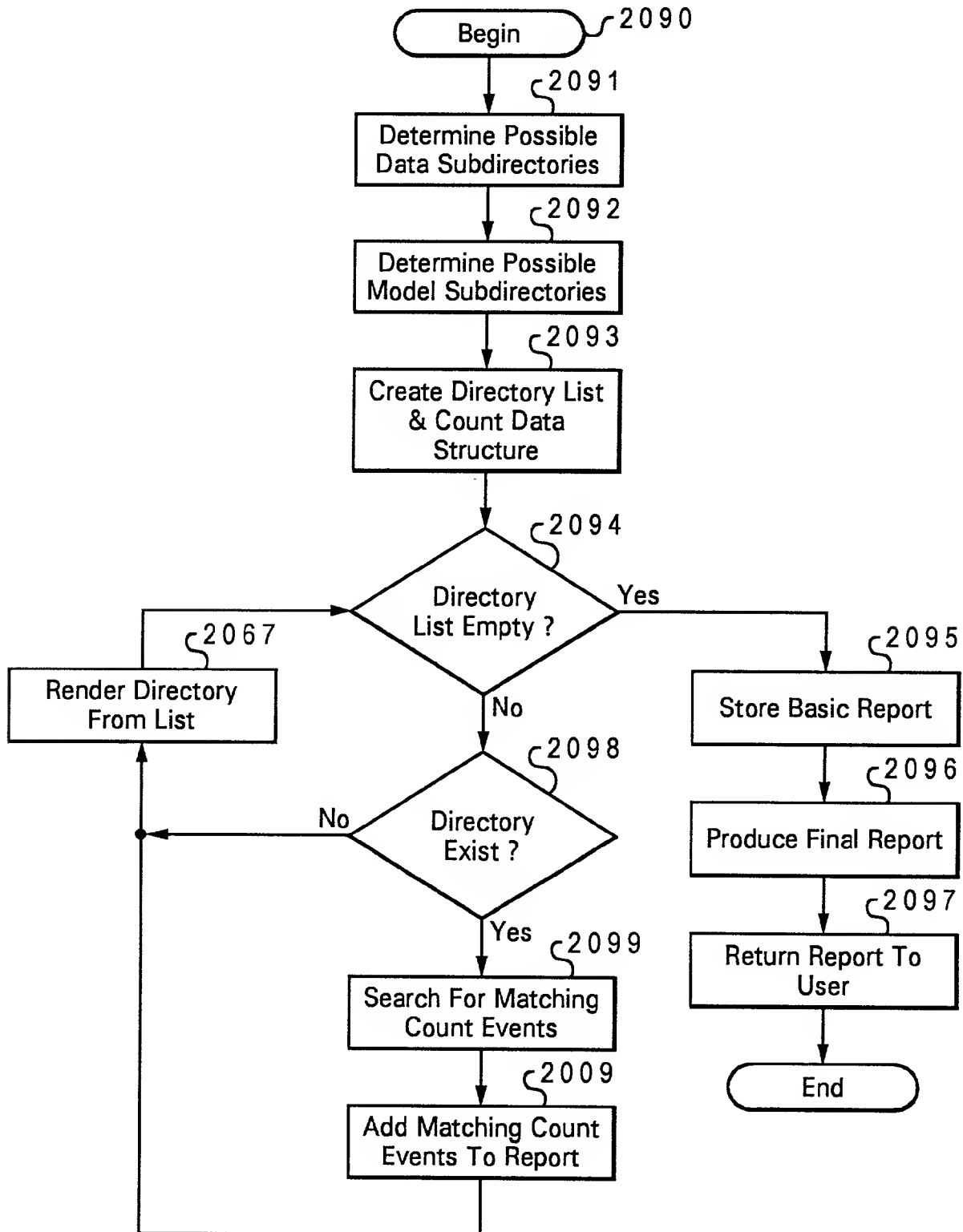


Fig. 20I

53/62

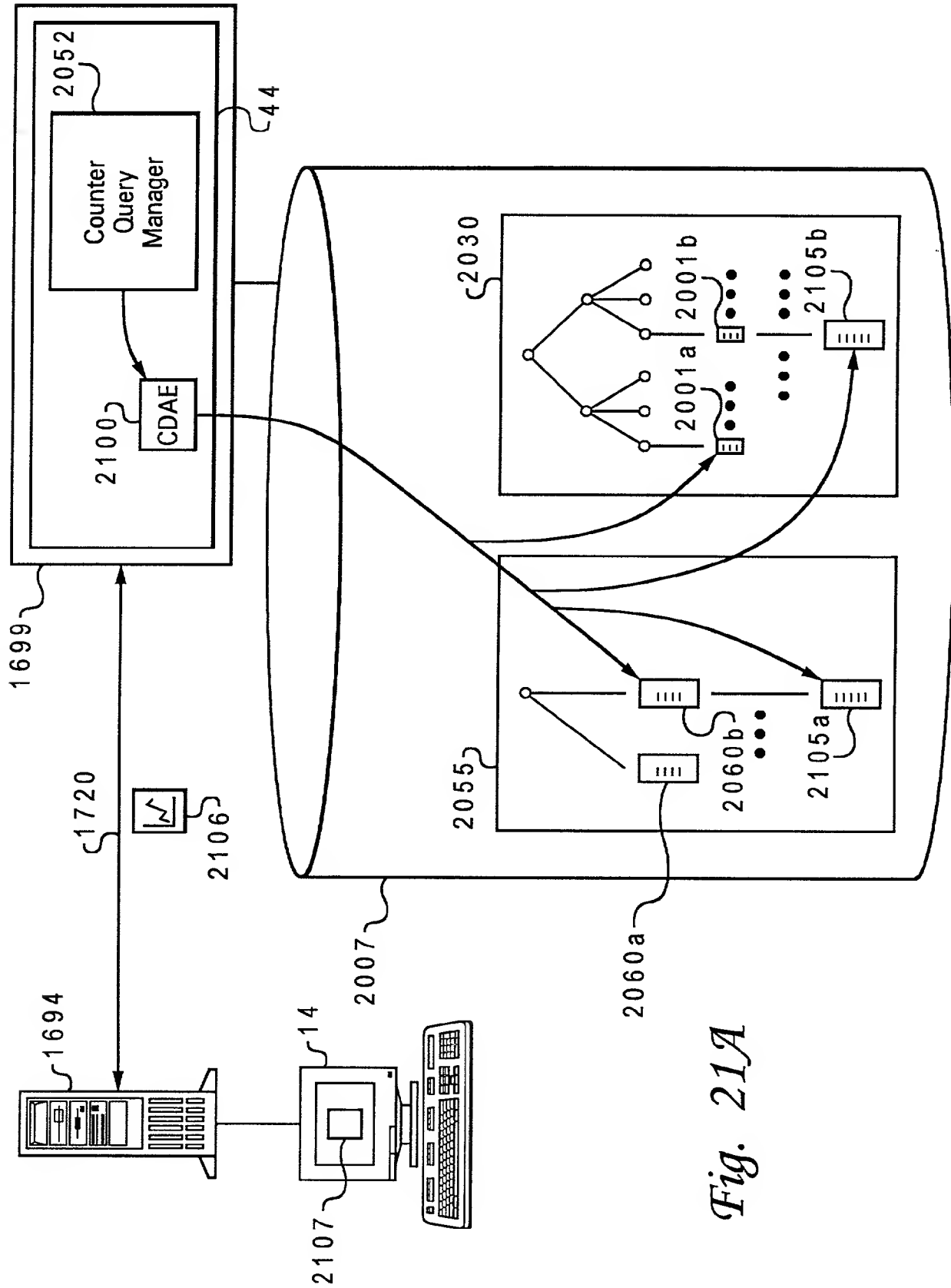


Fig. 21A

54/62

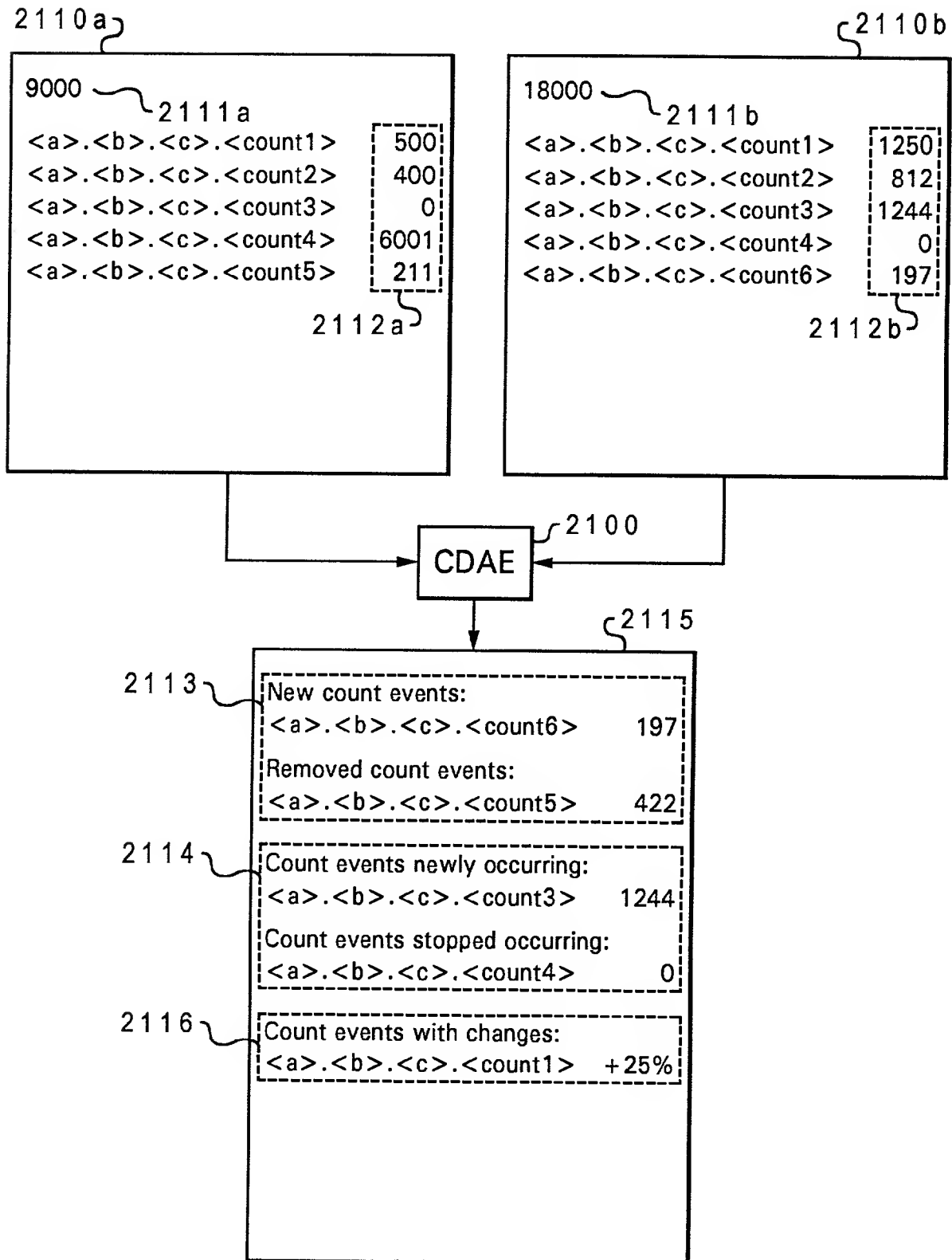


Fig. 21B

55/62

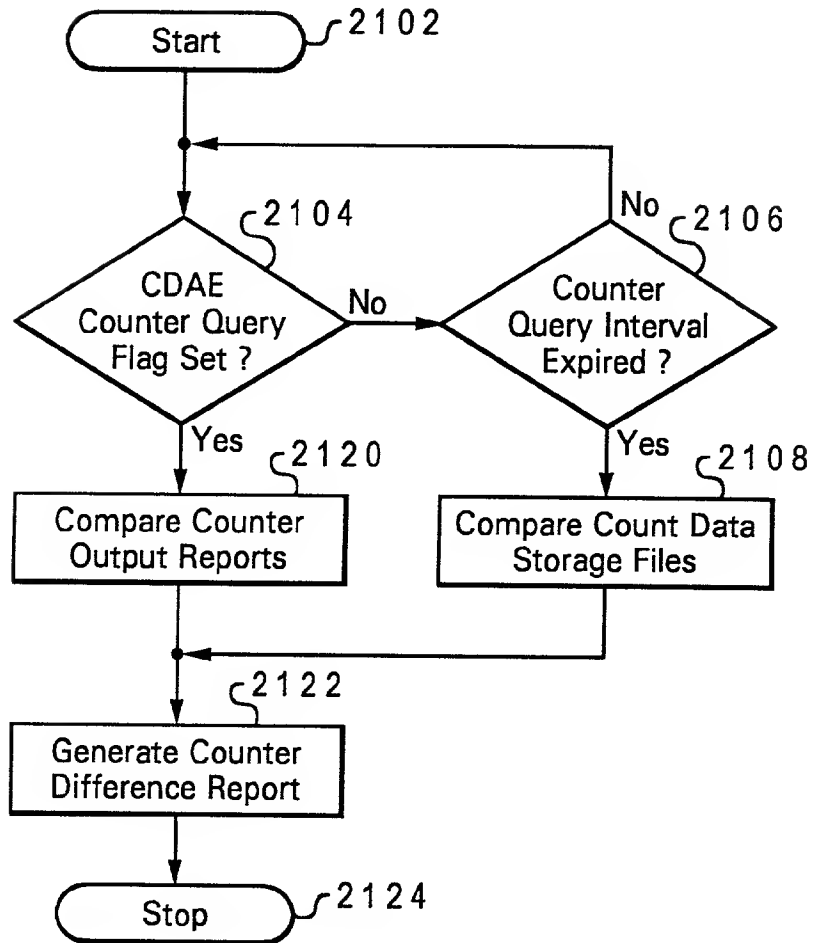
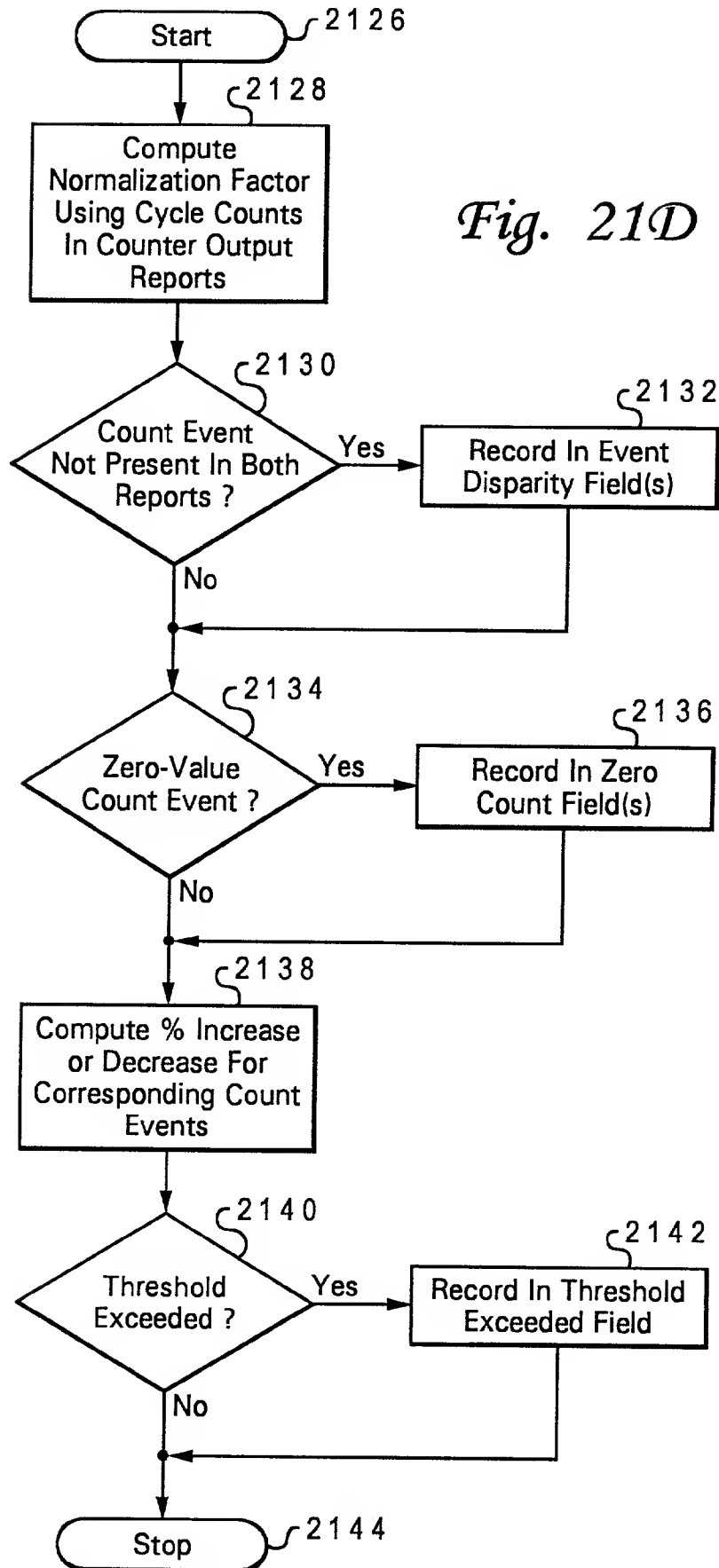


Fig. 21C



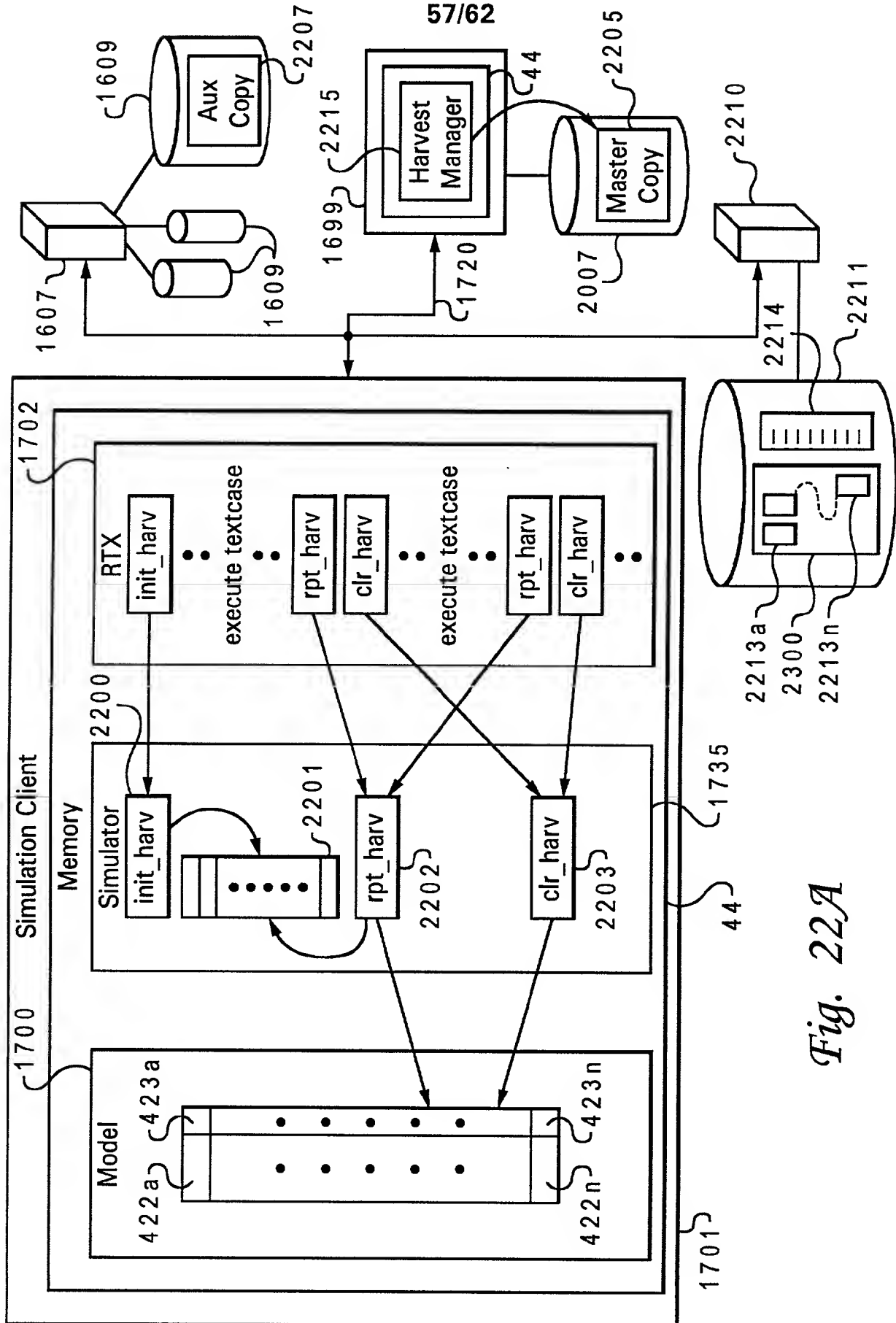


Fig. 22A

58/62

Fig. 22B

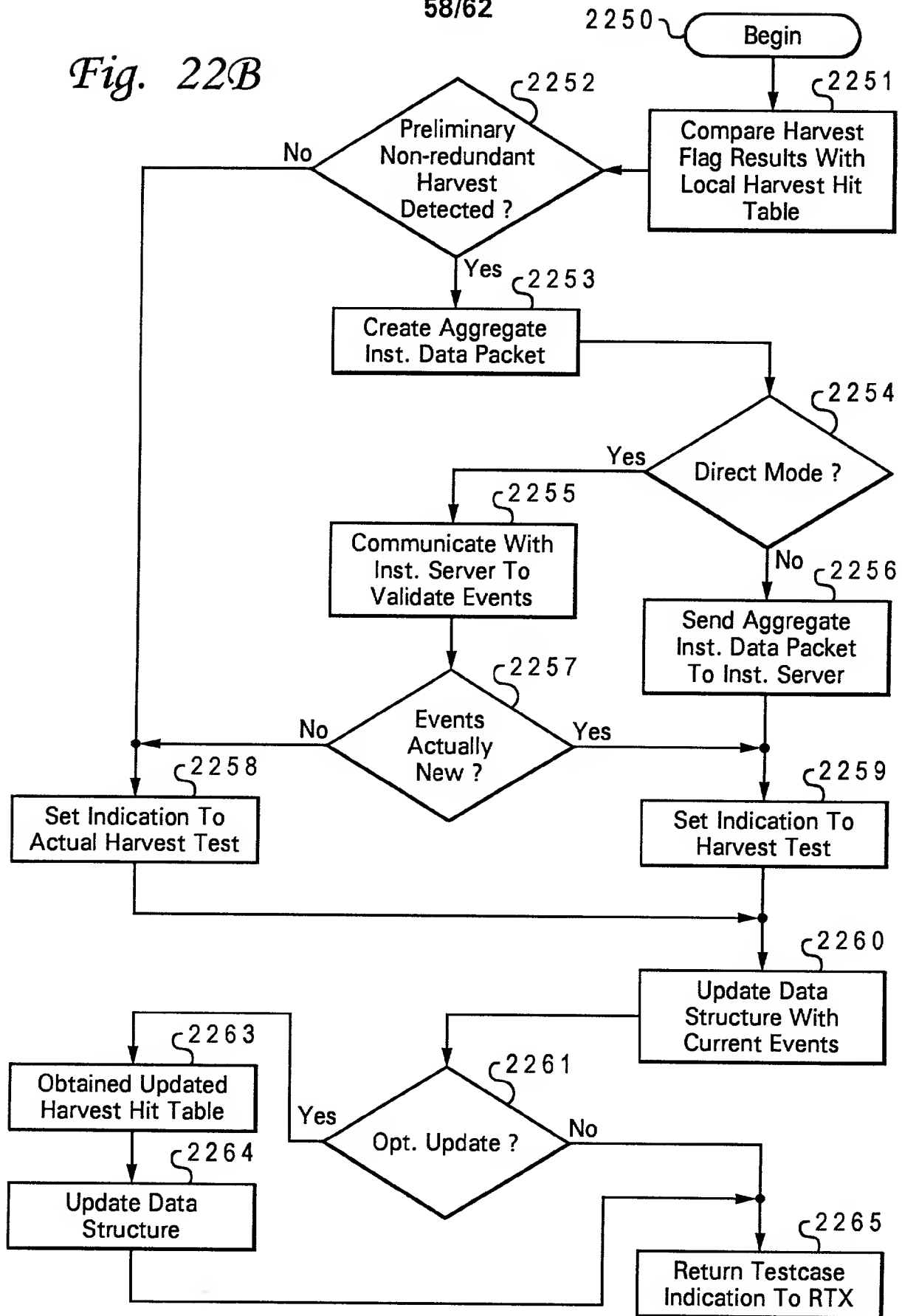
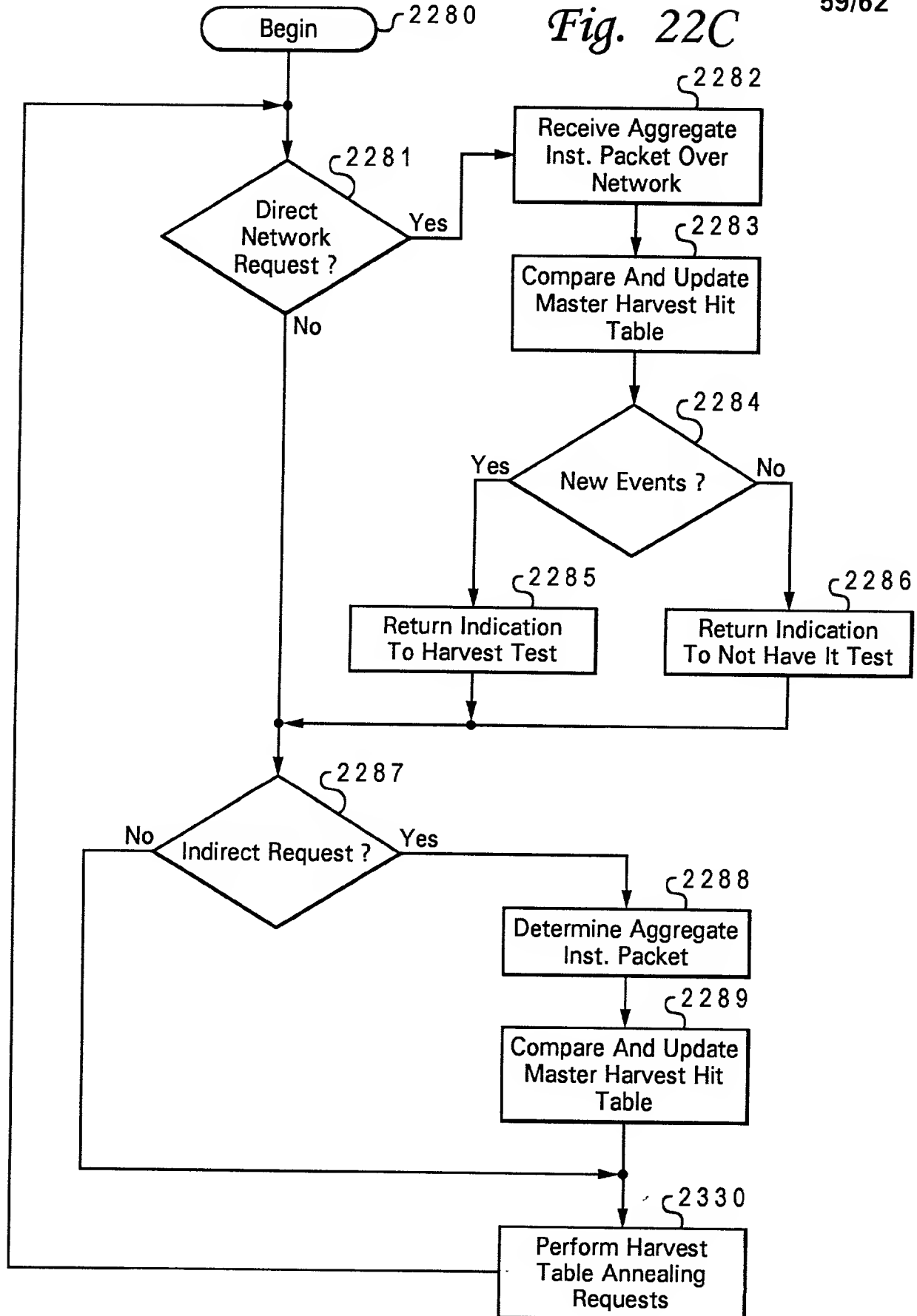


Fig. 22C



60/62

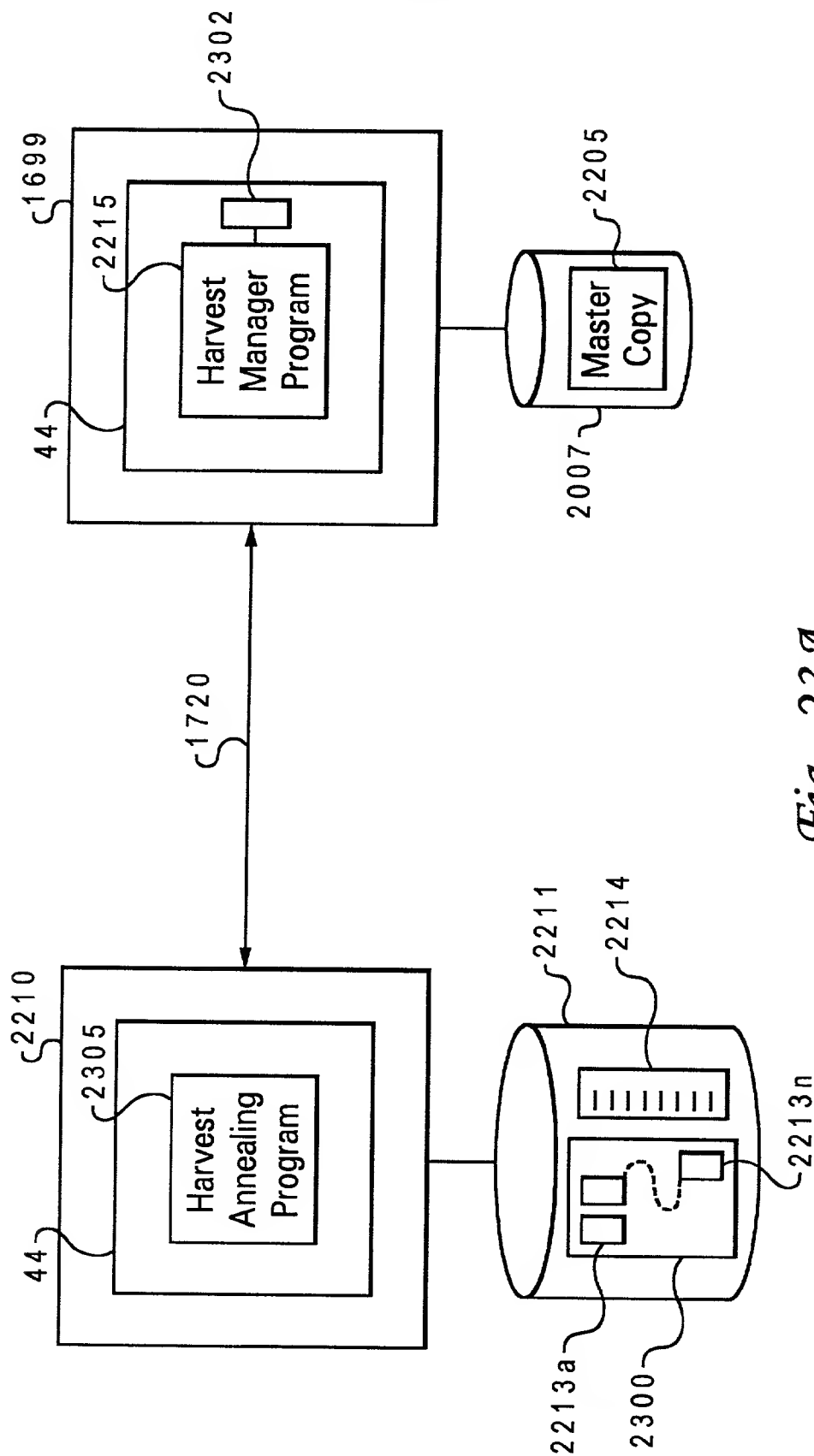


Fig. 23A

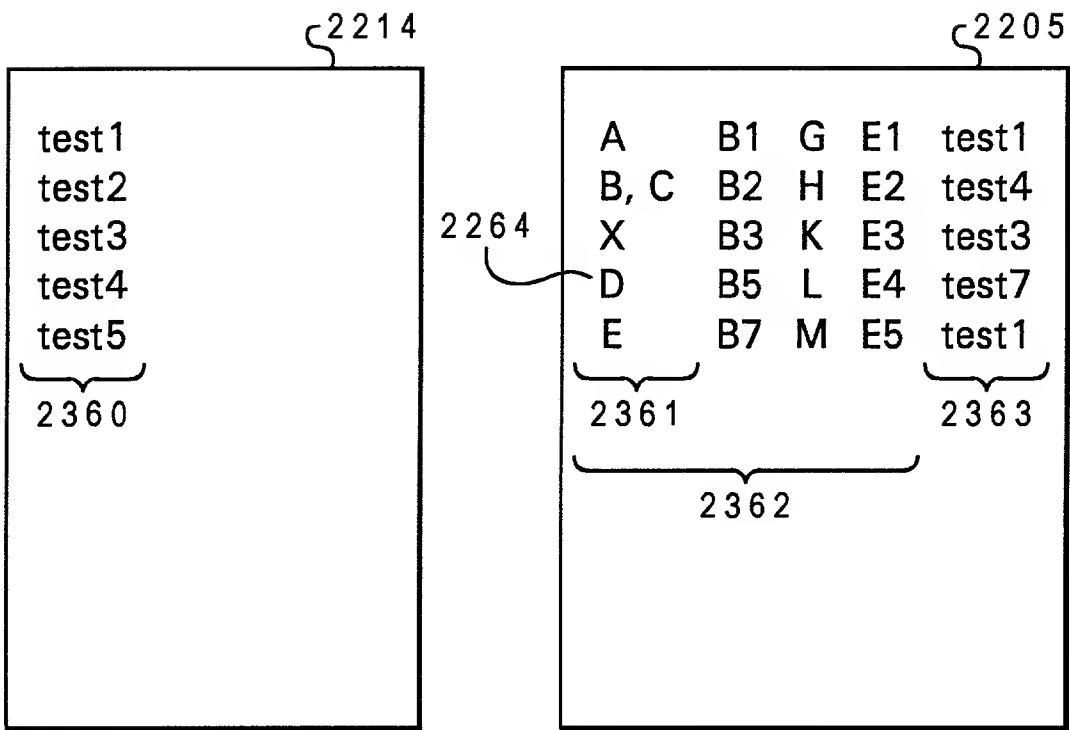


Fig. 23B

62/62

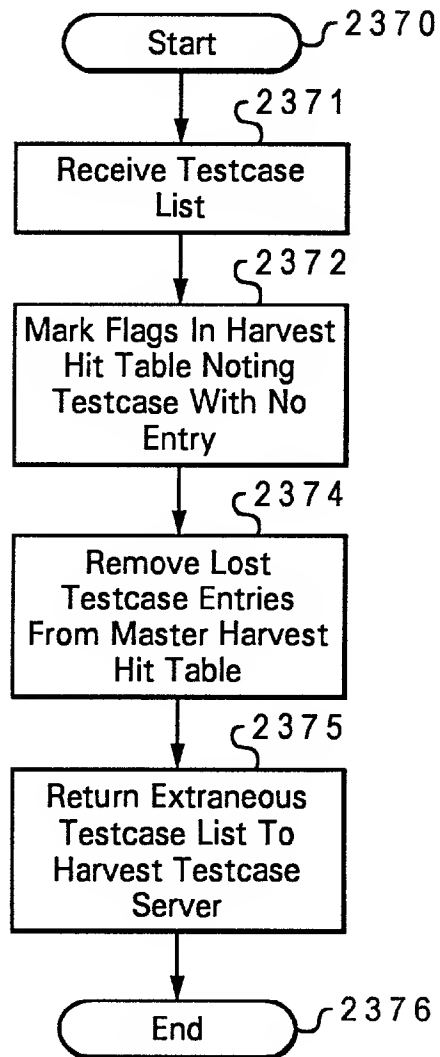


Fig. 23C